JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

VET CARE APPLICATION

Submitted by
ANAKAPALLI PRAKASH
Regd.No:321233620003
Under the Esteemed Guidance of
K. RATNAREDDY
ASSISTANT PROFESSOR

ABSTRACT

"Vet Care" is an application which is simple and easy to use for online prescription from veterinary doctor. This application is useful for distant areas, instant consulting and prescription. The proposed application provide system which is useful for layman. "Vet Care" is an android application which is useful for both doctor and users. Farmer has to send his animal's signs and symptoms. The doctor will receive the farmer's requests and responds. The user's data will be automatically added to the database. Farmer has to register first and has to subscribe. Farmer will freely subscribe to Govt. CVHs. He has to pay fee if he will subscribe to the private clinic's doctor. There will be two payment gateways for the users, bank transfer (Doctor will provide his bank information while creating his profile). Doctors will provide their bank details for payments. He can also provide easy paisa account details. There will be information of influenced animals with basic diseases and their prescriptions. On the basis of recorded data, the doctor can analysis that which disease attacks on vet and sugges the pescription.

CHAPTER 1 – INTRODUCTION

There are so many common animal diseases. Everybody doesn’t feel good to go to the doctor at an initial stage of any basic disease of his pet/animal. Even they do not know the basic treatment of the disease. In most of the cases, People bring their pets to the doctor at the middle stage of the disease.

“Vet Care” is an android application which is useful for both doctor and farmers.

Farmer has to send his animal’s signs and symptoms to doctor in video, audio, images or text form. The doctor will receive the farmer’s requests and responds.

1.1Scope:

“Vet Care” is an application which is simple and easy to use. Farmers that are registered can send the symptoms of his pet/animal to the doctor and get responds. This application is useful for distant areas. Farmer has to register first and has to subscribe. Farmer will freely subscribe to Govt. CVHs. The user has to pay fee if user will subscribe to the private clinic’s doctor. There will be images of influenced animals with basic diseases and their prescriptions.

1.1 Objectives

This company will be in service to all ranges of animals, from big to small. Our clients can feel comfortable with their beloved pets in our hands. All of our staff is well trained and happy to help. We feel that having a veterinary clinic that you can trust is having a happy pet’s health you can trust. We treat every pet patient as our own and guarantee safe service. The reason for starting this business is to help all pets in our area live happy, healthy lives. All of us at Beaver Creek Veterinary Clinic love animals and are trained and ready to take care of any problem that comes along. We believe that the focus of veterinary service and medicine is the bond between our clients and their pets. Our objectives of this business are:

- To provide care and protection from disease
- Help all of our patients live a long and healthy life
- To provide any advice that a client may have about their pet

- To provide an affordable full time medical service to all who are in need
- To provide more than medical help for pets such as boarding, tags, toys, and supplies

1.2 Purpose of proposed work

We are developing a system which is useful for layman. "Vet Care" is an android application which is useful for both doctor and farmers. Farmer has to send his animal's signs and symptoms to doctor in text form. The doctor will receive the farmer's requests and responds. The user's data will be automatically added to the database. Farmer has to register first and has to subscribe. Farmer will freely subscribe to Govt. CVHs. He has to pay fee if he will subscribe to the private clinic's doctor. There will be two payment gateways for the farmers, bank transfer (Doctor will provide his bank information while creating his profile) and easy paisa. Doctors will provide their bank details for payments. He can also provide easy paisa account details. There will be images of influenced animals with basic diseases and their prescriptions. On the basis of recorded data, the system can perform analysis that which disease attacks on pets mostly in a specific area.

1.3 System Specifications

Hardware Requirements:-

Processor	:	Intel(R) 2.10GHz
Installed memory (RAM)	:	4 GB
Hard Disk	:	160 GB
Operating System	:	Windows (7)

Software Requirements: -

Front End: HTML5, CSS3, Bootstrap

Back End: PHP 7.4, MYSQL

Control End: Angular Java Script

Tools:

xampp-win 64-8.1

CHAPTER 2 – LITERATURE REVIEW

VCMA literature reviews are science-based, peer-reviewed literature summaries of select topics relevant to animal health and welfare. They are written by VCMA professional staff in response to a demonstrated need for summary information. They may also include attention to the issue's legislative, legal, economic, ethical, and social components, as well as implications for veterinary practice. VCMA literature reviews are not VCMA policy, and generally do not draw conclusions; they simply report what we know from the literature or other verifiable data about a given topic. Peer-reviewers for VCMA literature reviews are selected by staff from among recognized experts in their field, after reviewing related peer-reviewed published works and seeking input from those working on the topic of interest. Recognizing that even science is not value-free, in addition to ensuring we involve the needed experts, we attempt to be comprehensive and inclusive with respect to reviewers' backgrounds and perspectives when soliciting comments. Peer-review for literature reviews is similar to peer-review used for publication of articles in scientific and other professional journals.

VCMA welcomes new information that will improve the content of its literature reviews. Please share any comments or suggestions you have on our literature reviews by sending them, with supporting documentation.

2.1 Existing Definition

- Currently there is no such system available in the market.
- The existing system is manual.
- Farmers have to bring their pets to the clinic for their treatment.
- Everybody doesn't feel good to take his pet to the doctor at initial stage.
- It's time consuming.
- Not in reach of distant people.

2.2 Proposed Solution:

We are developing a system which is useful for layman. "Vet Care" is an android application which is useful for both doctor and farmers. Farmer has to send his animal's signs and symptoms to doctor in text form. The doctor will receive the farmer's requests and responds. The user's data will be automatically added to the database. Farmer has to register first and has to subscribe. Farmer will freely subscribe to Govt. CVHs. He has to pay fee if he will subscribe to the private clinic's doctor. There will be two payment gateways for the farmers, bank transfer (Doctor will provide his bank information while creating his profile) and easy paisa. Doctors will provide their bank details for payments. He can also provide easy paisa account details. There will be images of influenced animals with basic diseases and their prescriptions. On the basis of recorded data, the system can perform analysis that which disease attacks on pets mostly in a specific area.

CHAPTER 3 OVERALL DESCRIPTION OF THE PROPOSED SYSTEM

3.1 Product Perspective

A simple, easy-to-use application designed to post pet's complaint details via this application with your Doctors. User first register to login this app. After enter the login and post their pet's complaint, symptoms, age, photo and etc.

After doctor enter the login and view pet's complaint and prepare for solutions. Then if they need for doctor consultation that detail also posted through this application. Finally user got this replay for solution after pay this consultation payment via this application.

Vet Care Mobile App using simple steps given below:

Step1: Start register for login to application

Step2: Enter login and click on to Continue

Step3: Select complaint details and post their complaint via this application.

Step4: Doctor enters login and view user's complaints and prepare solution for this treatment details. And post that report to user.

Step5: Finally user got reply for solution after enter the payment details.

3.2 Interfaces

The application will have a user friendly and menu based interface. Following user interface pages will be provided.

- Registration to have their detailed information to maintain database
- Login to validate and allow them to access these application complaints with already stored database.
- Doctor enters the login and view pet's complaint and prepare for solutions.
- Then if they need for doctor consultation that detail also posted through this application.

3.3 System Modules

Farmer has to send his animal's signs and symptoms to doctor in text form. He has to pay fee if he will subscribe to the private clinic's doctor.

System Modules:

USER

- ➢ Register
- ➢ Login
- ➢ Post Complaint
- ➢ My Complaint Status
- ➢ View Vetcure details
- ➢ View News
- ➢ View Doctor
- ➢ View Report
- ➢ Post Feedback
- ➢ Make Payment
- ➢ My Profile

Doctor

- ➢ Register
- ➢ Login
- ➢ View Complaint
- ➢ Post Solution
- ➢ Manage Cure

- Manage news
- View user Details
- View Payment Details
- Feedback Details

Module Description

Login Module

The main activities in the application are the user login page for user. The other modules are followed by this login page. This module records only user and password of the user.

Registration Module

Another main function of our proposed system is registration, in order to register with the unique application details such as bridal name, groom name; password, email, place and time are required.

Post Complaint

User posts their complaint through this application. It means pet's disease like symptoms, age, gender, photos those details are posted via this system.

Doctor Response

Doctor enters this system and view user complaint and gives solution for user complaint. And if they need doctor consultation those details also attach this replay.

Payment Details

Finally user got solution report for complaint while enter the payment for this consultation.

Cure Details

Admin can manage the diease cure details like disease, Symptom and remedies for the cure. User can search for cure details and get solution easily.

News Details

Admin can publish the news updates like vaccination or camp for animal. So that it user can view all details of new updates.

Feedback Registry

Admin can view all feedback details response from the user details.

CHAPTER 4 – DESIGN

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software.

The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer's requirements into finished software or a system.

Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data.

4.1UML Diagrams:

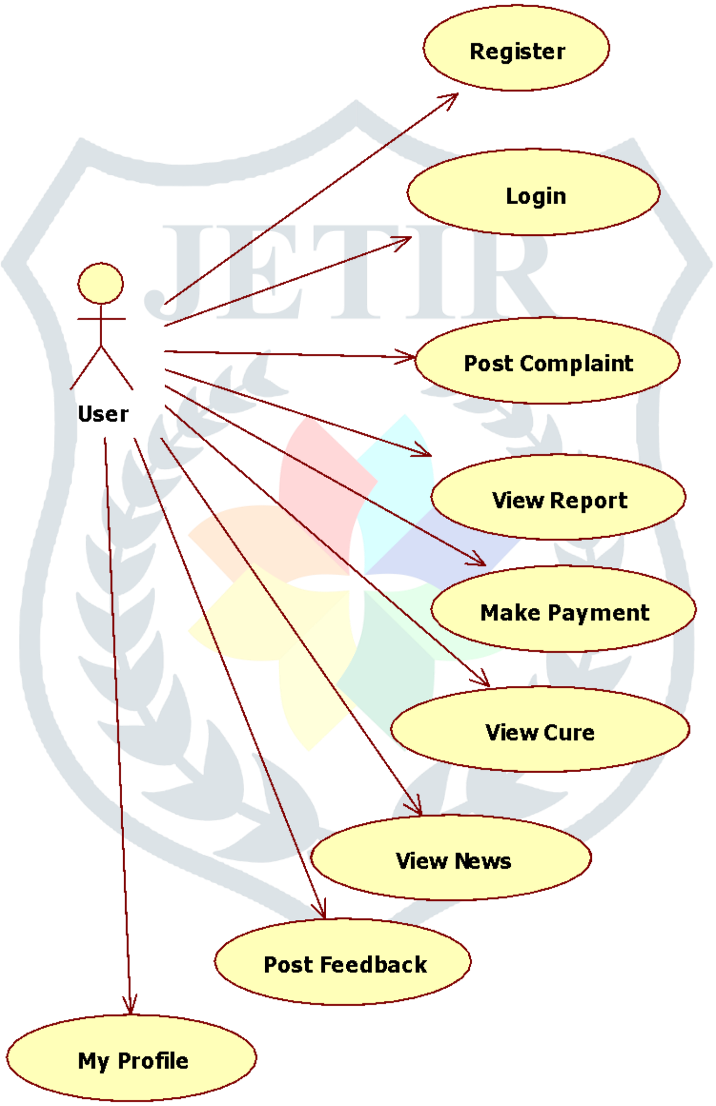
UML stands for Unified Modeling Language. UML is a language for specifying, visualizing and documenting the system. This is the step while developing any product after analysis. The goal from this is to produce a model of the entities involved in the project which later need to be built. The representation of the entities that are to be used in the product being developed need to be designed.

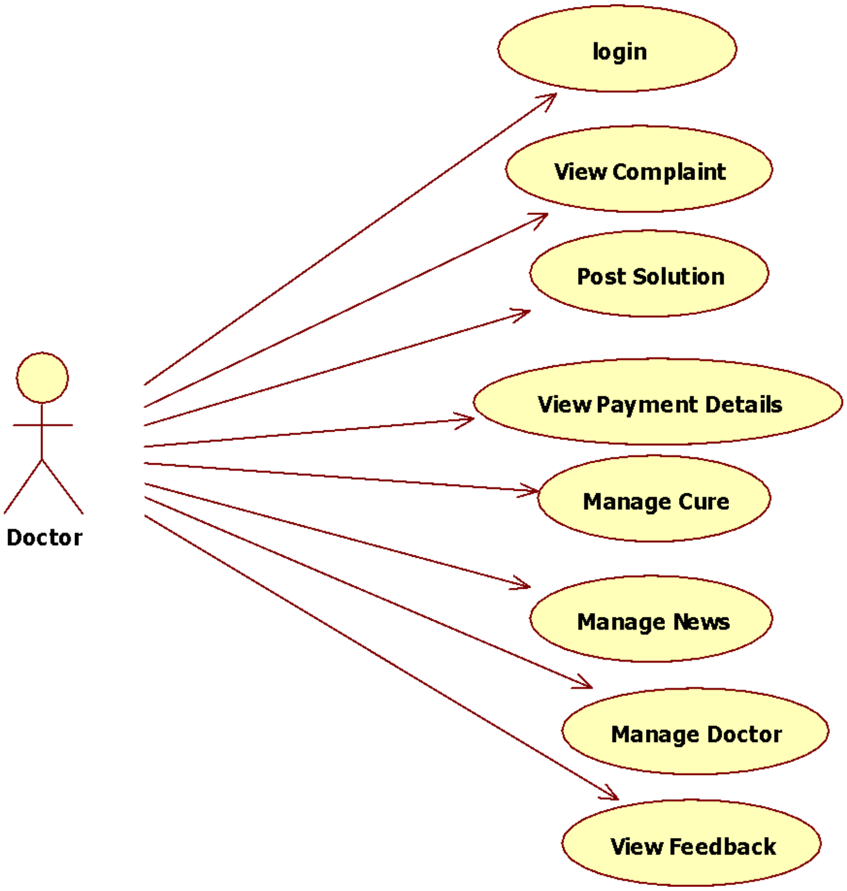
There are various kinds of methods in software design:

- Use case Diagram
- Sequence Diagram
- Class Diagram
- Activity Diagram.

4.1.1 Usecase Diagrams:

Use case diagrams model behavior within a system and helps the developers understand of what the user require. The stick man represents what's called an actor. Use case diagram can be useful for getting an overall view of the system and clarifying who can do and more importantly what they can't do.





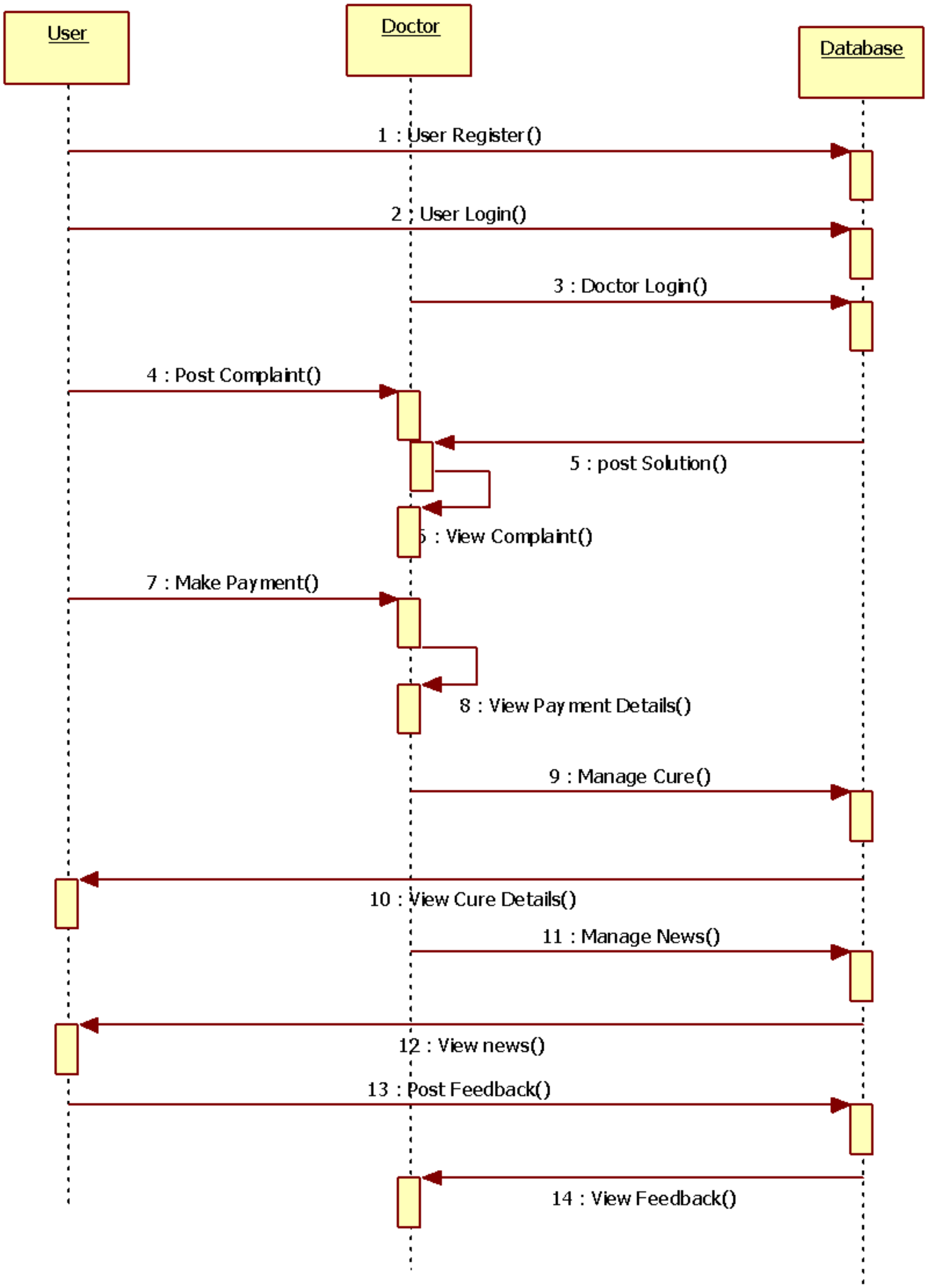
Use case diagram consists of use cases and actors and shows the interaction between the use case and actors.

- The purpose is to show the interactions between the use case and actor.
- To represent the system requirements from user's perspective.
- An actor could be the end-user of the system or an external system.

4.1.2 Sequence Diagram:

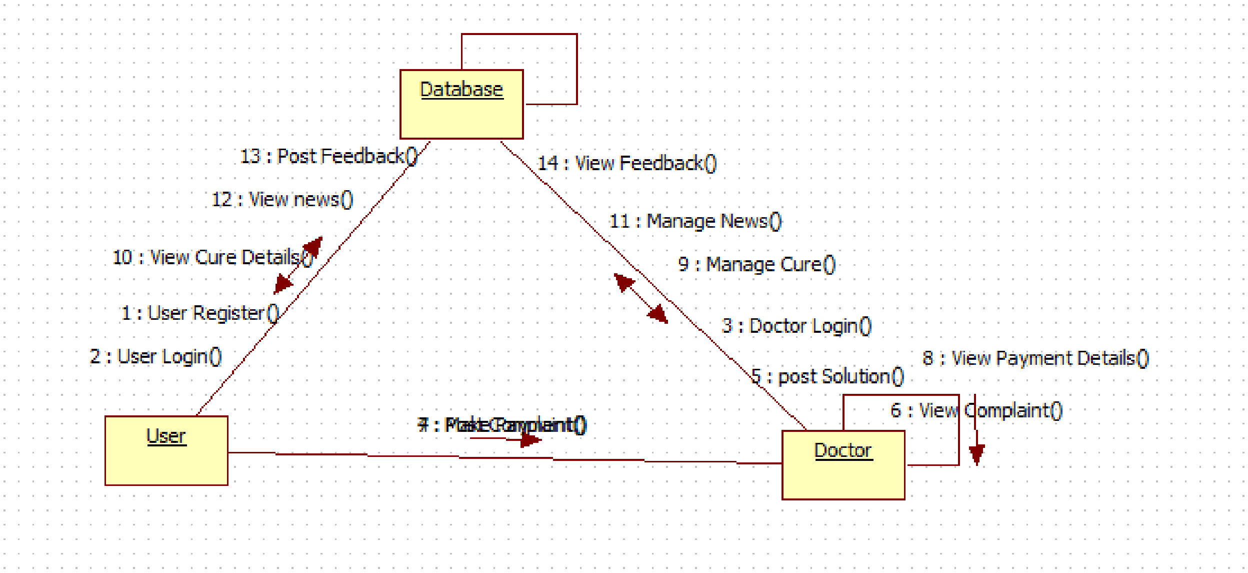
Sequence diagram and collaboration diagram are called INTERACTION DIAGRAMS. An interaction diagram shows an interaction, consisting of set of objects and their relationship including the messages that may be dispatched among them.

A sequence diagram is an introduction that empathizes the time ordering of messages. Graphically a sequence diagram is a table that shows objects arranged along the X-axis and messages ordered in increasing time along the Y-axis.



4.1.3 Collaboration Diagram:

A **collaboration diagram** is a type of visual presentation that shows how various software objects interact with each other within an overall IT architecture and how users can benefit from this **collaboration**. A **collaboration diagram** often comes in the form of a visual chart that resembles a flow chart.


Database
13 : Post Feedback()
12 : View news()
10 : View Cure Details()
1 : User Register()
2 : User Login()
14 : View Feedback()
11 : Manage News()
9 : Manage Cure()
3 : Doctor Login()
5 : post Solution()
8 : View Payment Details()
6 : View Complaint()
7 : Make Payment()
4 : Post Complaint()
User
Doctor

JETIR

4.1.4 Table Design

User Register & Login

User ID	Name	Email Id	Password	Mobile
Int	Varchar	Varchar	Varchar	Varchar
100	100	100	100	100
Primary key				

Admin Login

User ID	Name	Email Id	Password	Mobile
Int	Varchar	Varchar	Varchar	Varchar
100	100	100	100	100
Primary key				

Complaint Table

User ID	Date	Animal Type	Age	Category	Weight	No of days suffering	Suffering From	Symptons	Status
Int	Varchar	Varchar	Varchar	Varchar	Varchar	Varchar	Varchar	Varchar	Varchar
100	100	100	100	100	100	100	100	100	100
Primary key									

Cure

ID	Disease	Symptons	Soultion	Process	Description	Notes
Int	Varchar	Varchar	Varchar	Varchar	Varchar	Varchar
100	100	100	100	100	100	100
Primary key						

News

ID	News Topic	Description
Int	Varchar	Varchar
100	100	100
Primary key		

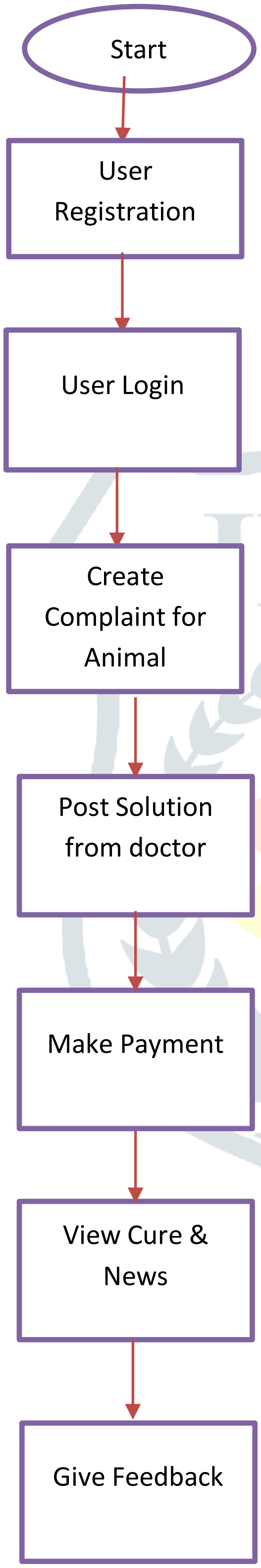
Feedback

User ID	Name	Email	Feedback
Int	Varchar	Varchar	Varchar
100	100	100	100
Primary key			

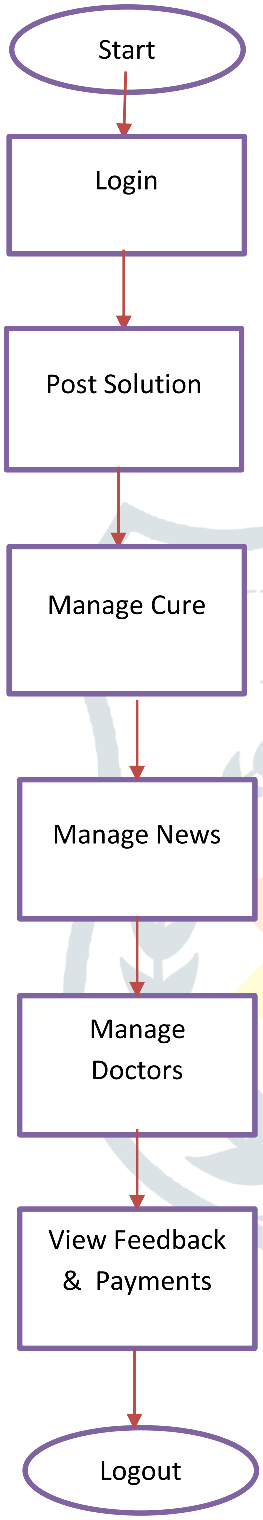
JETIR

4.1.5. Data Flow Diagram

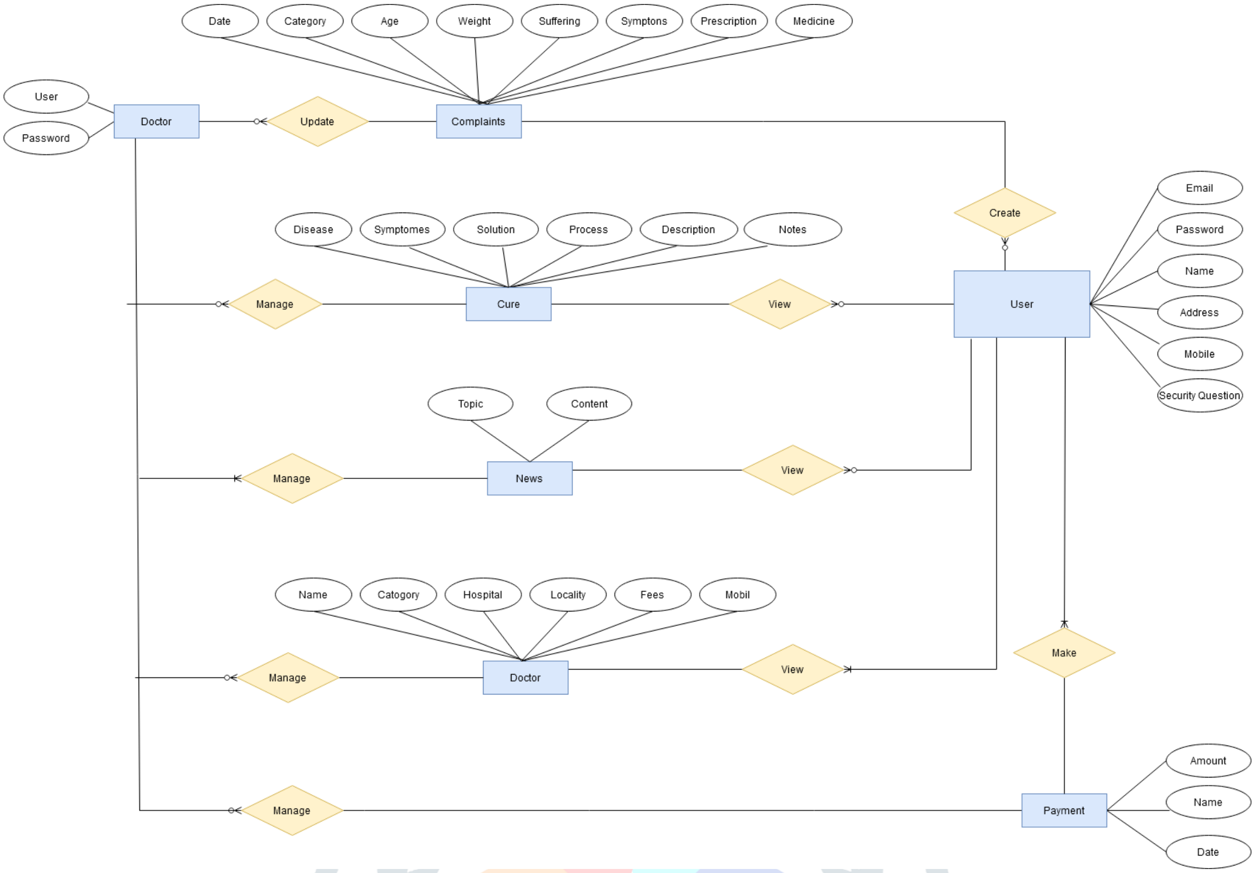
User module



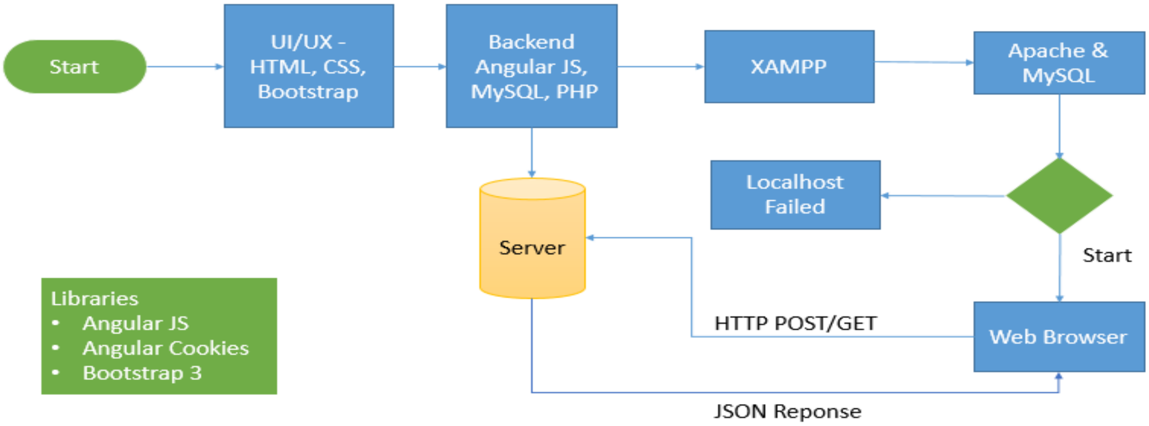
Admin module



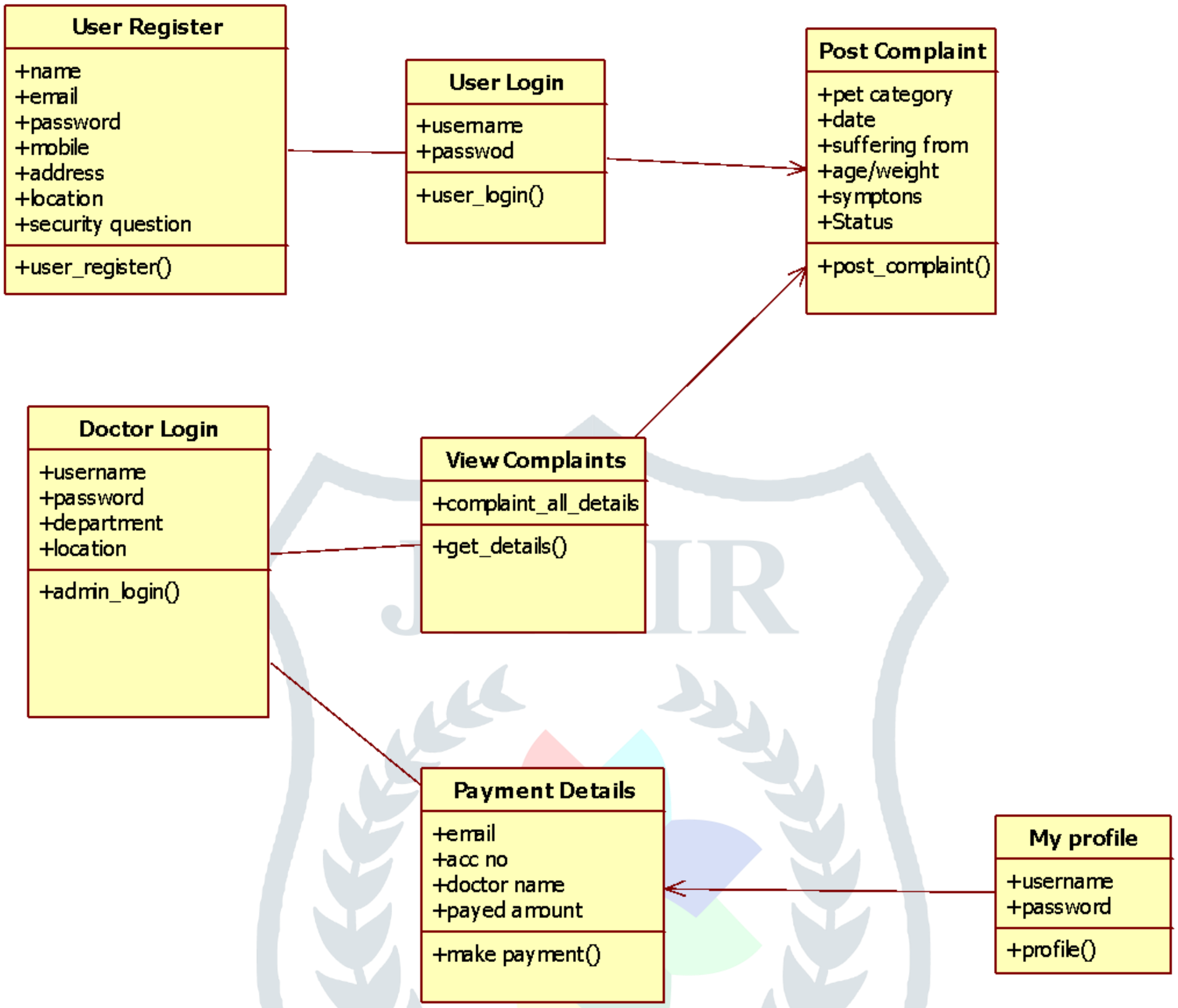
4.1.6 ER Diagram:



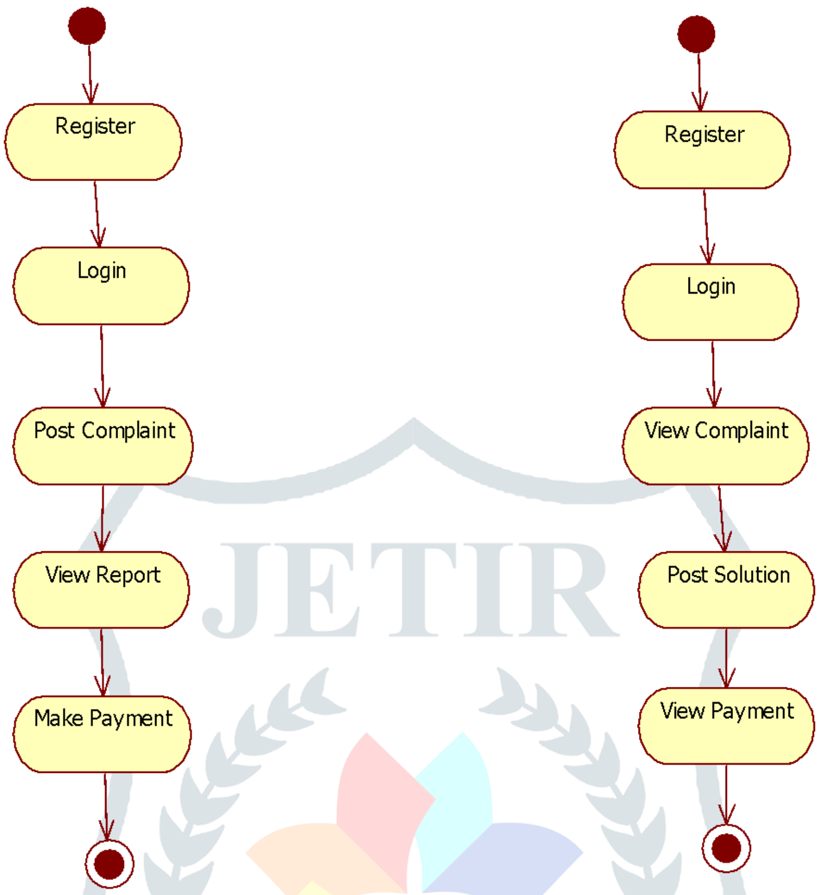
4.1.7 Workflow Diagram:



4.1.8 Class Diagram:

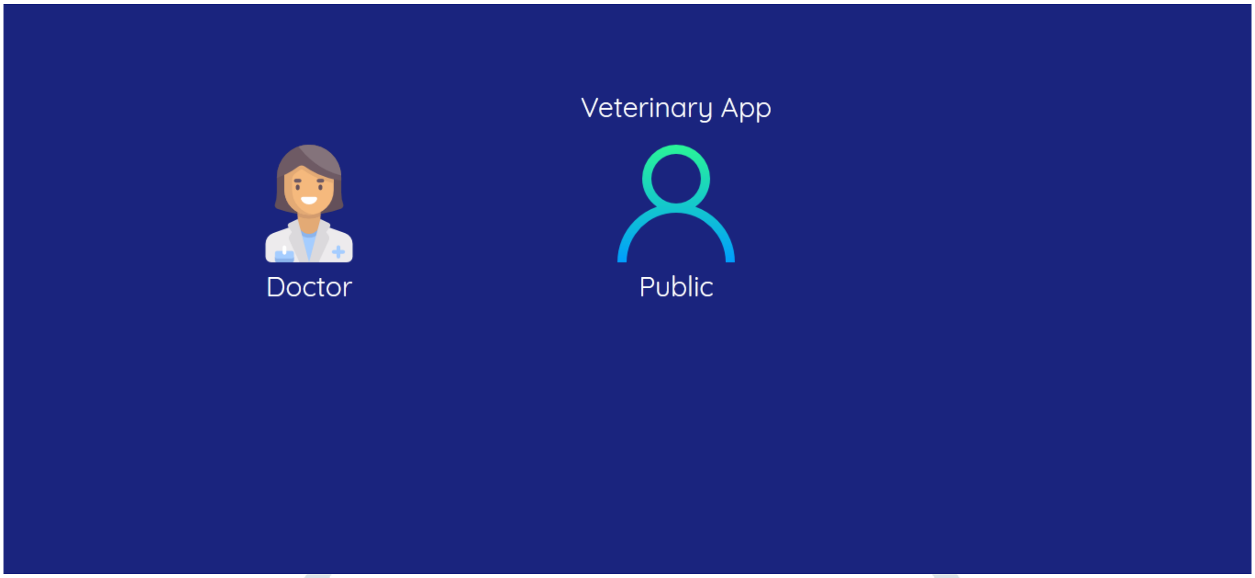


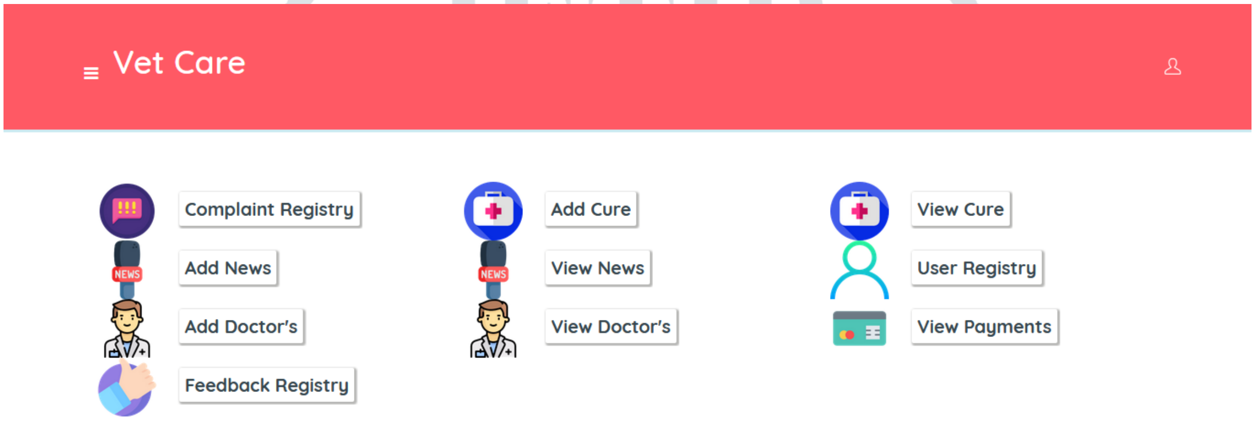
4.1.9 Activity Diagram:

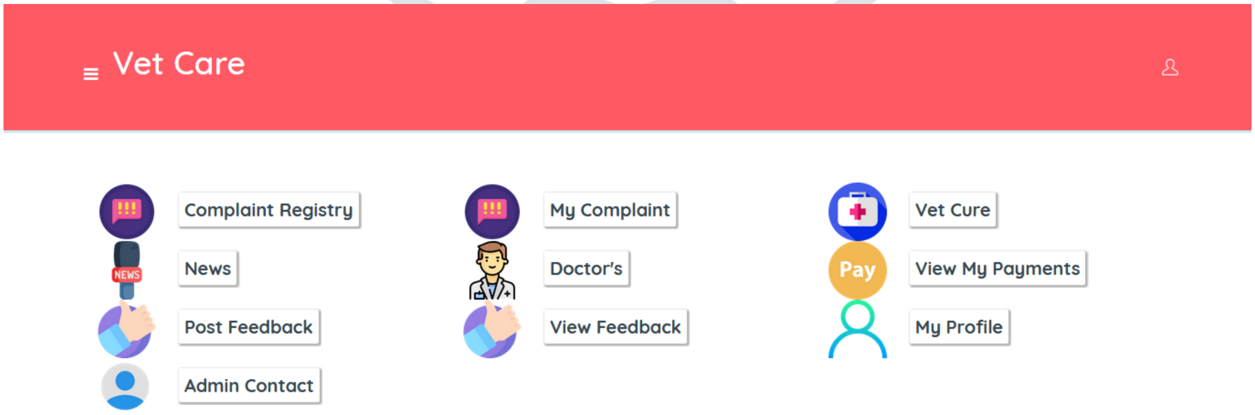


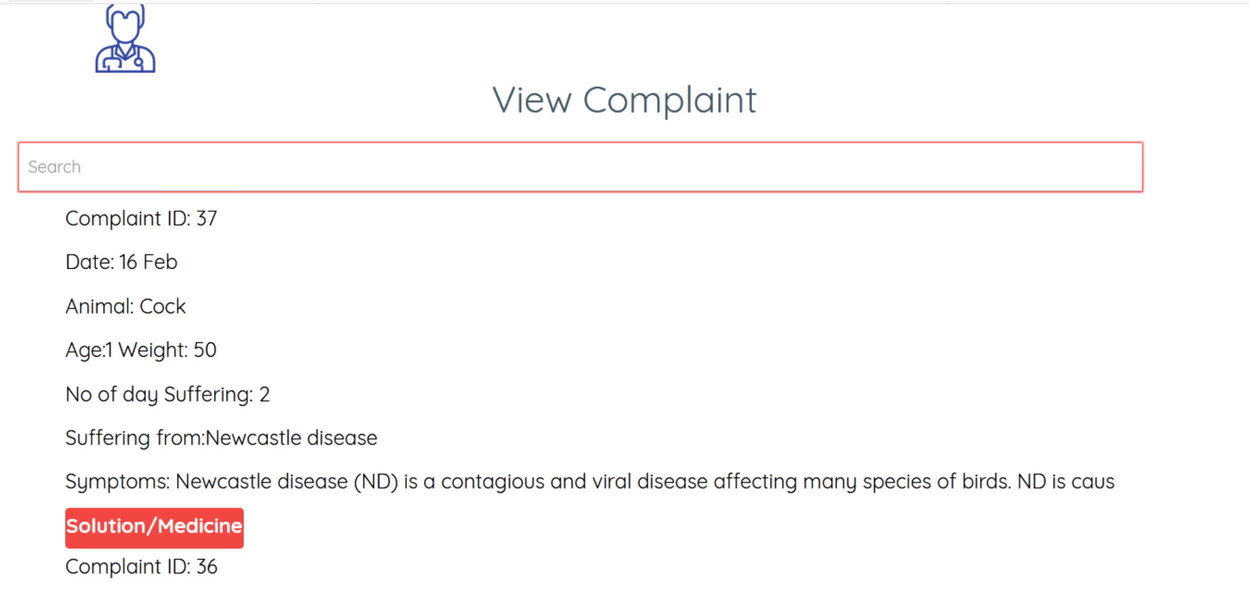
JETIR

CHAPTER 5 - OUTPUT SCREENSHOTS









View Complaint

Complaint ID: 37

Date: 16 Feb

Animal: Cock

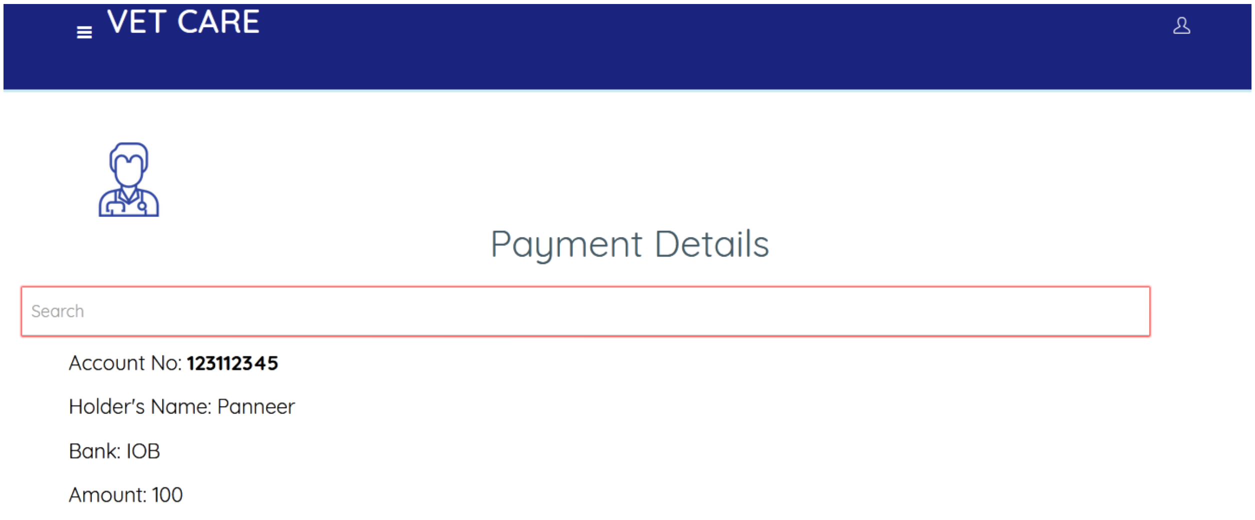
Age:1 Weight: 50

No of day Suffering: 2

Suffering from:Newcastle disease

Symptoms: Newcastle disease (ND) is a contagious and viral disease affecting many species of birds. ND is caus

Complaint ID: 36



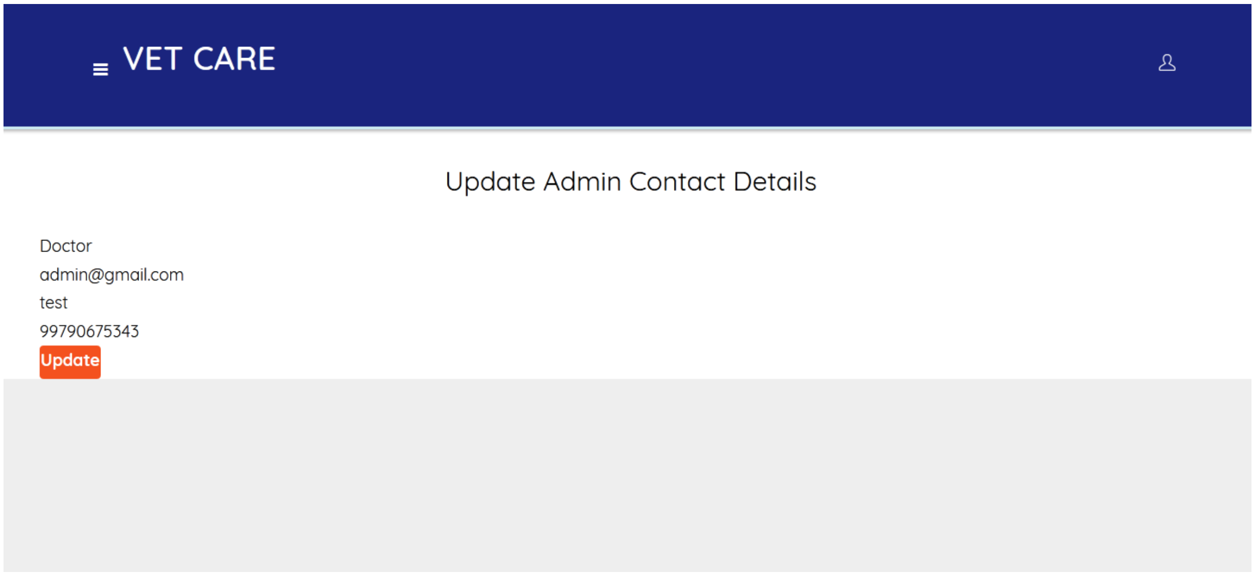
Payment Details

Account No: **123112345**

Holder's Name: Panneer

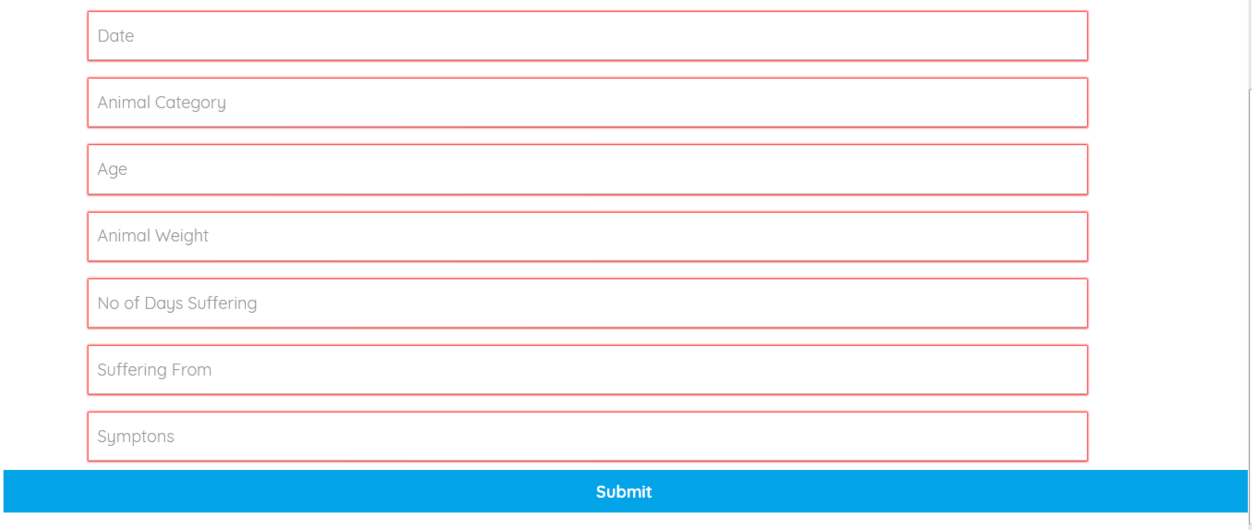
Bank: IOB

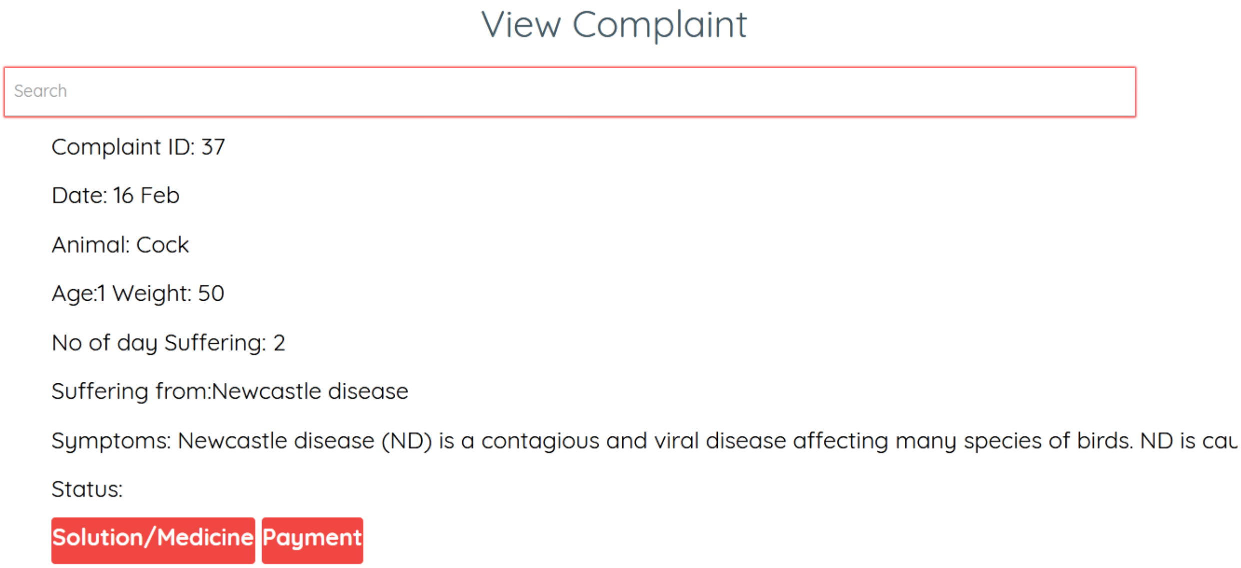
Amount: 100



Update Admin Contact Details

Doctor
admin@gmail.com
test
99790675343





Complaint ID: 37

Date: 16 Feb

Animal: Cock

Age:1 Weight: 50

No of day Suffering: 2

Suffering from:Newcastle disease

Symptoms: Newcastle disease (ND) is a contagious and viral disease affecting many species of birds. ND is cau

Status:

CHAPTER 6 – IMPLEMENTATION DETAILS

6.1 Introduction to Html Framework

HyperText Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages. Along with CSS, and JavaScript, HTML is a cornerstone technology used to create web pages, as well as to create user interfaces for mobile and web applications. Web browsers can read HTML files and render them into visible or audible web pages. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language.

HTML elements form the building blocks of HTML pages. HTML allows images and other objects to be embedded and it can be used to create interactive forms. It provides a means to create structured documents by denoting structuralsemantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated bytags, written using angle brackets. Tags such as <img /> and <input /> introduce content into the page directly. Others such as <p>...</p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages. HTML markup can also refer the browser to Cascading Style Sheets (CSS) to define the look and layout of text and other material

6.2 Cascading Style Sheets (CSS)

CSS is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG andXUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content, such as semantically insignificant tables that were widely used to format pages before consistent CSS rendering was available in all major browsers. CSS makes it possible to separate presentation instructions from the HTML content in a separate file or style section of the HTML file. For each matching HTML element, it provides a list of formatting instructions. For example, a CSS rule might specify that "all heading 1 elements should be bold", leaving pure semantic HTML markup that asserts "this text is a level 1 heading" without formatting code such as a<bold> tag indicating how such text should be displayed.

This separation of formatting and content makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser orscreen reader) and on Braille-based, tactile devices. It can also be used to display the web page differently depending on the screen size or device on which it is being viewed. Although the author of a web page typically links to a CSS file within the markup file, readers can specify a different style sheet, such as a CSS file stored on their own computer, to override the one the author has specified. If the author or the reader did not link the document to a style sheet, the default style of the browser will be applied. Another advantage of CSS is that aesthetic changes to the graphic design of a document (or hundreds of documents) can be applied quickly and easily, by editing a few lines in one file, rather than by a laborious (and thus expensive) process of crawling over every document line by line, changing markup.

The CSS specification describes a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities (or weights) are calculated and assigned to rules, so that the results are predictable.

6.3 MYSQL Server

MySQL is an open-source relational database management system (RDBMS);[6] in July 2013, it was the world's second most widely used RDBMS, and the most widely used open-source client–server model RDBMS. It is named after co-founder Michael Widenius's daughter, My. The SQL acronym stands for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedishcompany MySQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality.

6.4 PHP

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now stands for therecursive backronym PHP: Hypertext Preprocessor.

PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management system and web frameworks. PHP code is usually processed by a PHPinterpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface(CLI) and can be used to implement standalone graphical applications.

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system andplatform, free of charge.

The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as a de facto standard. Since 2014 work has gone on to create a formal PHP specification.

6.5 ANGULAR JAVA SCRIPT

AngularJS (commonly referred to as "Angular" or "Angular.js") is an open-source web application framework mainly maintained by Google and by a community of individuals and corporations to address many of the challenges encountered in developing single-page applications. It aims to simplify both the development and the testing of such applications by providing a framework for client-side model–view–controller (MVC) and model–view–viewmodel(MVVM) architectures, along with components commonly used in rich Internet applications.

The AngularJS framework works by first reading the HTML page, which has embedded into it additional custom tag attributes. Angular interprets those attributes as directives to bind input or output parts of the page to a model that is represented by standard JavaScript variables. The values of those JavaScript variables can be manually set within the code, or retrieved from static or dynamic JSON resources.

According to JavaScript analytics service Libscore, AngularJS is used on the websites of Wolfram Alpha, NBC,Walgreens, Intel, Sprint, ABC News, and approximately 8,400 other sites out of 1 million tested in July 2015.

AngularJS is the frontend part of the MEAN stack, consisting of MongoDB database, Express.js web application server framework, Angular.js itself, and Node.js runtime environment.

7. SYSTEM STUDY

7.1 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER 8 - SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TESTCASE

S.NO	SCENARIO	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT
1	User register	Email and Password	All the user details register successfully	Register successfully Or Register unsuccessfully
2	User Login	Email and Password	If correct directed to home page otherwise show "Invalid Login"	Login successfully or Login unsuccessfully
3	Post complaint	User will enter all post complaint	If all the post complaint "Created Successfully"	Created successfully or created unsuccessfully
4	My Complaint Status	User check and view all Complaint status details	user view Complaint Status details	View My Complaint Status details
5	View Vetcure Details	User check and view all Vetcure Details	User view Vetcure Details	View division Vetcure Details
6	View News	User check and view all News Details	User view News Details	View division News Details
7	View Doctor	User check and view all Doctor Details	User view Doctor Details	View division Doctor Details
8	View Report	User check and view all Report	User view Report Details	View division Report Details

9	Post Feedback	User will enter all post Feedback	If all the post Feedback “Created Successfully”	Created successfully or created unsuccessfully
10	Make Payment	User add the Payment Details	all the Payment details “created successfully”	created successfully or unsuccessfully
11	My Profile	User can edit and profile	If any changes or user can edit profile	Updated successfully or unsuccessfully
12	Doctor register	Email and Password	All the user details register successfully	Register successfully Or Register unsuccessfully
13	Doctor Login	Email and Password	If correct directed to home page otherwise show “Invalid Login”	Login successfully or Login unsuccessfully
14	View Complaint	User check and view all Complaint	User view Complaint Details	View division Complaint Details
15	Post Solution	User will enter all post Solution	If all the post Solution “Created Successfully”	Created successfully or created unsuccessfully
16	Manage Cure	Doctor add the Cure details	all the Cure details “updated successfully”	updated successfully or unsuccessfully
17	Manage News	Doctor add the News details	all the News details “updated successfully”	updated successfully or unsuccessfully
18	View User Details	Doctor check and view all User Details	Doctor view User Details	View division User Details

JETIR

19	View Payment Details	Doctor check and view all Payment Details	Doctor view Payment Details	View division Payment Details
20	View Feedback Details	Doctor check and view all Feedback Details	Doctor view Feedback Details	View division Feedback Details

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

8.1 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

8.2 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

8.3 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

CHAPTER 9– CONCLUSIONS

This is android base application. Easily accessible to the farmers. Currently available manual system is not more effective. Farmer can treat his pet/animal at the farm rather than bringing his pet to the clinic. The appropriate application of telemedicine can enhance animal care in rural communities by facilitating expedient communication, diagnostics, and possible treatment recommendations, client education improvement. Consumers so that they cannot just solve a crisis, but they can be better informed and thereby prevent medical crises.

CHAPTER 10- REFERENCES

1.Animal Health at the Crossroads: Preventing Detecting and Diagnosing Animal Diseases, National Research Council National Academies of Sciences, Engineering, and Medicine, pp. 30-75.

2.P.F. Cowhey and J.D. Aronson, Transforming Global Information and Communication Markets: The Political Economy of Innovation, Cambridge:MIT Press, 2009.

3.M. DeBell and C. Chapman, Computer and Internet use by students in 2003 (NCES 2006-065), Washington, DC:U.S. Department of Education, National Center for Educational, 2006.

4.Jason Keath, *Ultimate Recap: The 12 Best New Social Media Tools from 2012*, [online] Available: http://socialfresh.com/best-social-media-tools-2012/.

5.*Agricultural Statistics Board U.S. Department of Agriculture*, September 2007.

6.*Trend data: Adult gadget ownership over time*, 2012, [online] Available: http://pewinternet.org/Trend-Data-Adults/Device-Ownership.aspx.

7.M. Madden, A. Lenhart, M. Duggan, S. Cortesi and U. Gasser, *Teens and technology 2013. Pew Internet & American Life Project*, 2013, [online] Available: http://www.pewinternet.org.-vrnedia//Files/Reports/2013/PIP_TeensandTechnology2013.pdf.

8.C. Metzler, M. Sanders, J. Rusby and R. Crowley, "Using consumer preference information to increase the reach and impact of media-based parenting interventions in a public health approach to parenting support", *Behavior Therapy*, vol. 43, pp. 257-270, 2012.

9.L. Rainie and S. Fox, "Just-in-time information through mobile connections", *Pew Internet & American Life Project*, 2012, [online] Available: http://www.pewinternet.org/Reports/2012/Just-in-time.asp.

CHAPTER 11- CODING

ANGULAR_USER.JS

```
var app = angular.module("myapp", ['ngCookies']);

app.controller("myappCtrl", function($scope, $cookieStore, $cookies, $http)

{

        /************************** Cookies Login **************/

        $scope.cook_doctor_email = $cookieStore.get("cook_doctor_email");

        /*****************************************************************************/

/************************** Doctor Login *************************************/

/****************************************************************************/

        // sign in button

        $scope.doctor_login = function()

        {

    $http.post('doctor_login.php',

                        {'email': $scope.email, 'password':$scope.password})

                .success(function(data, status, headers, config)

                {

                        if(data.success == 1)

                        {

                                alert("Login Successful");

                                $cookieStore.put("cook_doctor_email",data.email);

                                window.location = "admin_home.html";  // Home Page
```

```
                                        return;

                                }

                                else if(data.success == 2)

                                {

                                        alert("Please Fill All Fields");

                                }

                                else

                                {

                                        alert("Login Unsuccessful");

                                }

          });

        }




/****************************************************************************/

/************************** User Login *************************************/

/****************************************************************************/

            // sign in button

            $scope.user_login = function()
```

```
  {

$http.post('user_login.php',

          {'email': $scope.email, 'password':$scope.password})

      .success(function(data, status, headers, config)

      {

          if(data.success == 1)

          {

              alert("Login Successful");

              $cookieStore.put("cook_user_email",data.email);

              window.location = "home.html";  // Home Page

              return;

          }

          else if(data.success == 2)

          {

              alert("Please Fill All Fields");

          }

          else

          {

              alert("Login Unsuccessful");

          }

});
```

```
    }



/************************* Cookies **********************************/

            $scope.cookie_user_email = $cookieStore.get("cook_user_email");




//************************* admin_register **********************************/


            $scope.user_register = function()

            {

                        $http.post('user_register.php',{

                         'name':$scope.name,'email':$scope.email,'password':$scope.password,

                         'mobile': $scope.mobile})

                        .success(function(data, status, headers, config)

                        {

                                    if(data.success == 1)

                                    {

                                                alert("Registered successfully");

                                                window.location = "user_login.html";

                                                return;

                                    }
                        else

                        {

                                alert("Invalid Inputs");

                        }

      });

   }



/****************************************************************************/

/************************** Admin account Details View *********************************/

/****************************************************************************/

          $http.post('admin_contact_get.php')

          .success(function(data, status, headers, config)

          {

                  if(data.success == 1)

                  {

                          $scope.admin_account_details = data.details;

                  }

                  else

                  {

                          $scope.admin_account_details = "No Data Found !!!";

                  }
```

```
 });

/************************ Cookies Login ********************************/


        $http.post('get_admin_info.php')

        .success(function(data, status, headers, config)

        {

            if(data.success == 1)

            {

                $scope.details = data.details;

            }

   });




    $scope.myinfovar = true;



     $scope.update_info = function(name,email,password,mobile)

    {

        $scope.myinfovar = false;



        $scope.email = email;

        $scope.password = password;
```

```
                $scope.name = name;

                $scope.mobile = mobile;

                //window.location = "myprofile.html";

}



$scope.save_info = function()

{

                $http.post('admin_update.php',{

                 'name':$scope.name,'email':$scope.email,

                 'password':$scope.password,'mobile': $scope.mobile})

                .success(function(data, status, headers, config)

                {

                                if(data.success == 1)

                                {

                                                alert("Submited successfully");

                                                window.location = "user_update_info.html";

                                                return;

                                }

                                else

                                {

                                                alert("Invalid Inputs");
```

```
            }

  });

}



    /**************************************************************************/

    $scope.user_logout = function()

    {

        if(confirm("Are You Sure?"))

        {

            $cookies.cook_user_email = "";

            $cookies.cook_admin_email = "";

            window.location = "index.html";

            return;

        }

        else

        {

            return false;

        }

    }



    /************************* Update User Info **********************************/
```

```
                $http.post('get_user_info.php',

                {

                        'email':$scope.cook_user_email

                })

                .success(function(data, status, headers, config)

                {

                                $scope.userdetails = data.details;

      });




});
```

USER_LOGIN.PHP

```
<?php

/* Following code will match admin login credentials */



//user temp array

$response = array();



// include db connect class
```

```
require_once __DIR__ . '/db_connect.php';


// connecting to db

$db = new DB_CONNECT();


// check for post data

$data = json_decode(file_get_contents("php://input"));

$get_empid = mysql_real_escape_string($data->email);

$get_password = mysql_real_escape_string($data->password);


if(empty($get_empid) || empty($get_password))

{

        $response["success"] = 2;

        echo json_encode($response);

}

else

{

        $result = mysql_query("SELECT * FROM login WHERE email = '$get_empid' AND password = '$get_password'");


                if (mysql_num_rows($result))

                {
                    $Allresponse = mysql_fetch_array($result);

                    // temp user array

                    $response = array();

                    $response = $Allresponse;

                    $response["success"] = 1;

                    echo json_encode($response);

          }

          else

          {

                    // success

                    $response["success"] = 0;

                    // echoing JSON response

                    echo json_encode($response);

          }

}

?>
```

INDEX.HTML

```
<!DOCTYPE html>

<html ng-app="myapp">

          <head>
```

```
<!-- Basic -->
<meta charset="utf-8">
<meta name="keywords" content="HTML5 Template" />
<meta name="description" >
<meta name="author" content="pixelgeeklab.com">

<!-- Mobile Metas -->
<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!-- Web Fonts  -->
<link href='css/css.css' rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="fonts/font.css">

<!-- Vendor CSS -->
<link rel="stylesheet" href="vendor/fontawesome/css/font-awesome.css">
<link rel="stylesheet" href="vendor/owlcarousel/owl.carousel.css" media="screen">
<link rel="stylesheet" href="vendor/owlcarousel/owl.theme.css" media="screen">
<link href="vendor/owl-carousel/owl.transitions.html" rel="stylesheet" media="screen">
<link rel="stylesheet" href="vendor/flexslider/flexslider.css" media="screen">
<link rel="stylesheet" href="vendor/chosen/chosen.css" media="screen">
<link rel="stylesheet" href="vendor/magnific-popup/magnific-popup.css" media="screen">
```

```
        <!-- Theme CSS -->

        <link rel="stylesheet" href="css/theme.css">

        <link rel="stylesheet" href="css/theme-animate.css">




        <!-- Style Switcher-->



        <!-- Head libs -->

        <script src="vendor/modernizr/modernizr.js"></script>



        <!--[if IE]>

                <link rel="stylesheet" href="css/ie.css">

        <![endif]-->



        <!--[if lte IE 8]>

                <script src="vendor/respond/respond.js"></script>

                <script src="vendor/excanvas/excanvas.js"></script>

        <![endif]-->



<style>
```

```
.container {

        background-color: #fa7881;

        position:relative;

        margin-top:100px;

}

.row{

margin-left:50px;

margin-right:50px;

text-align:center;

}

.login p{

position:relative;

text-align:center;

font-size:30px;

color:#ffffff;

}



p{

position:relative;

text-align:center;

font-family:'Quicksand';
font-size:35px;

color:#ffffff;

font-weight:200;

}

h2{

position:relative;

text-align:center;

font-size:30px;

color:#ffffff;

font-weight:200;

}

h3{

text-align:center;

font-size:80px;

color:#ffffff;

font-weight:200;

}

img{

align:middle;

}
```

```
    </style>


<!-- script back button -->

                <script src="cordova.js"></script>

                <script>

                function onLoad()

                {

                        document.addEventListener("deviceready", deviceReady, false);

                }

                function deviceReady()

                {

                        document.addEventListener("backbutton", backButtonCallback, false);

                }

                function backButtonCallback()

                {

                        navigator.app.exitApp();

                }

                </script>


        <!-- angular js -->

        <script src="js/angular-1.3.js"></script>
    <script src="js/angular_cookies.js"></script>



</head>

<body >

        <div class="container">





            <div class="row">

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

                <h2> Veterinary App</h2>

            </div>

                <div class="col-md-4">

                    <img src="images/doctor.png">

                        <div class="login">

                            <a href="doctor_login.html">

                            <p>Doctor</p>

                                </a>

                        </div>

                </div>

                <div class="col-md-4">

                    <img src="images/group.png">
                        <div class="login">
                            <a href="user_login.html">
                            <p>Public</p>
                            </a>
                        </div>
                </div>



            </div>
        </div>


        <script src="vendor/jquery/jquery.js"></script>
        <script src="vendor/bootstrap/bootstrap.js"></script>
        <script src="vendor/jquery.validation/jquery.validation.js"></script>
        <script src="vendor/owlcarousel/owl.carousel.js"></script>
        <script src="vendor/flexslider/jquery.flexslider-min.js"></script>
        <script src="vendor/countdown/countdown.min.js"></script>
        <script src="vendor/chosen/chosen.jquery.min.js"></script>
        <script src="vendor/pricefilter/jquery.pricefilter.js"></script>
        <script src="vendor/masonry/imagesloaded.pkgd.min.js"></script>
```

```
        <script src="vendor/masonry/masonry.pkgd.min.js"></script>

        <script src="vendor/uikit/uikit.js"></script>

        <script src="vendor/magnific-popup/jquery.magnific-popup.js"></script>



        <!-- Theme Base, Components and Settings -->

        <script src="js/theme.js"></script>



        <!-- Style Switcher -->

        <script type="text/javascript" src="style-switcher/js/switcher.js"></script>



    <!-- angular js -->

    <script src="js/angular_user.js"></script>

    <body onload="onLoad()">

</body>

</html>
```