FULL STACK PROJECT REPORT

**On**

**“Animal Haven Website”**

**Submitted by**

**Prachi Agrawal**

**(191500552)**

**Ritika Brewal**

**(191500663)**

**Priya**

**(191500599)**

**Neha**

**(191500489)**

Department of Computer Engineering & Applications

**Institute of Engineering & Technology**



**GLA University**

**Mathura- 281406, INDIA**

**2020-2021**

**Declaration**

We hereby declare that the work which is being presented in the Full Stack Project “**ANIMAL HAVEN WEBSITE”,** in partial fulfillment of the requirements for Full Stack Project viva voce, is an authentic record of our own work carried by the team members under the supervision of our mentor Mr. Pankaj Kapoor.

Group Members: Prachi Agrawal(191500552)

Ritika Brewal(191500663)

Priya (191500599)

Neha(191500489)

Course: B.Tech (Computer Science and Engineering)

Year: 3rd

Semester: 5th

Supervised By:

Mr. Pankaj Kapoor, Assistant Professor,

GLA University, Department of Computer Engineering & Application

**Certificate**

This is to certify that the above statements made by the candidates are correct to the best of my/our knowledge and belief.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Supervisor

Mr. Pankaj Kapoor

Technical Trainer

Dept of CEA, GLA University

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Project Coordinator Program Coordinator

(Mr. Mayank Srivastava) (Mr. Shashi Shekar)

**About the Project**

Our full stack project “Animal Haven Website” is an online website which will provide a location for Veterinarians hospitals in a city using one’s current location that will be very helpful for treatment of injured animals on time.. It consists of lists of vetenary hospitals in various city of India. List consists of hospital names and locations. Users login with their google accounts for the website that is designed and maintained by the social-media organization. Social media helps the development of online social networks by connecting a user's profile with those of other individuals or groups.

**Motivation**

We see injured animals on road, but we are unable to provide them treatment due to lack of knowledge of Veterinaries in the area.So, through this project we will be sharing locations of veterinaries in the area or near that area and people who want to help animals can easily took injured animals to hospitals. Not only website provides the location but also other services like emergency services.

**Requirements**

**a). Software Requirements:**

* Technology Implemented: Full Stack Web Development
* Languages/Technologies Used: HTML, CSS3, JavaScript ,Bootstrap
* IDE Used: Visual Studio Code
* Web Browser: Google Chrome
* GitHub: GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere. GitHub Repository: A GitHub repository can be used to store a development project. It can contain folders and any type of files (HTML, CSS, JavaScript, Documents, Data, Images). A GitHub repository should also include a license file and a README file about the project. A GitHub repository can also be used to store ideas, or any resources that you want to share.
* Visual Studio Code: Visual Studio Code is a free source-code editor made by Microsoft for Windows, Linux and macOS. [7] Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. Microsoft has released Visual Studio Code's source code on the VS Code repository of GitHub.com, under the permissive MIT License, while the compiled binaries are freeware.

**b). Hardware Requirements:**

* Processor Required: Intel i5
* Operating System: Windows 10
* RAM: 8GB
* Hardware Devices: Computer System
* Hard Disk: 256GB

**Acknowledgement**

We thank the almighty for giving us the courage and perseverance in completing the project. This project itself is an acknowledgement for all those people who have given us their heartfelt co-operation in making this project a grand success. We extend our sincere thanks to Mr. Pankaj Kapoor, Assistant Professor at “GLA University, Mathura” for providing his valuable guidance at every stage of this project work. We are profoundly grateful towards the unmatched services rendered by him. And last but not least, we would like to express our deep sense of gratitude and earnest thanks giving to our dear parents for their moral support and heartfelt cooperation in doing the main project.

**ANIMAL HAVEN WEBSITE**

**Abstract**

As the name suggests, our project is all about a helping injured animals. On roads, nearby, we see many animals roaming here and there and when they crosses road(generally), they hitted with vehicles and get injured an sometimes injury can be so serious that, that animal die on the moment. But if there is chance that animals can be saved by helping them, providing them proper and on-time treatment, we can save animals. We already see various species of animals are loosing, we need tos save animals. But let assume the situation, when an animal is injured by the person, who dont have idea how to treat animals or where to took the animal for treatment, in that case either the person waste the time on finding vetenaries (and delay in treatment may cause death of animals!). Instead of asking to peoples or wasting time on google map, people share their locations and according to their locations, website help to show nearest hospitals.

**Contents**

**Acknowledgment…………………………………….........**08

**Abstract…………………………………………………**09

1. **Chapter-1**

* **Introduction:**

Introduction to FULLSTACK........…......12-13

Pre-requisites…………………………..14

**2: Chapter-2…………………………………………………15-20**

**3. List of Figures...……………………………………**20-26

**4. Software Testing..............................................................**27-31

**5.Conclusion……………………………………………**32

**6.Bibliography………………………………………….**33

**Chapter 1**

**Introduction**

Today Developers around the world are making efforts to enhance user experience of using application as well as to enhance the developer’s workflow of designing applications to deliver projects and rollout change requests under strict timeline. Stacks can be used to build web applications in the shortest span of time. The stacks used in web development are basically the response of software engineers to current demands. They have essentially adopted pre-existing frameworks (including JavaScript) to make their lives easier. While there are many, MEAN and MERN are just two of the popular stacks that have evolved out of JavaScript. Both stacks are made up of open-source components and offer an end-to-end framework for building comprehensive web apps that enable browsers to connect with databases. The common theme between the two is JavaScript and this is also the key benefit of using either stack. One can basically avoid any syntax errors or any confusion by just coding in one programming language, JavaScript. The languages used are

 – Html , CSS , JavaScript ,Bootstrap

FULL STACK

A full stack web developer is a person who can develop both **client** and **server** software.

In addition to mastering HTML and CSS, he/she also knows how to:

**Front End and Back End:** Frontend and Backend are the two most popular terms used in web development. These terms are very crucial for web development but are quite different from each other. Each side needs to communicate and operate effectively with the other as a single unit to improve the website’s functionality.

**1: Front End**

**Front End Development:** The part of a website that the user interacts with directly is termed the front end. It is also referred to as the ‘client side’ of the application. It includes everything that users experience directly: text colors and styles, images, graphs and tables, buttons, colors, and navigation menu. HTML, CSS, and JavaScript are the languages used for Front End development. The structure, design, behavior, and content of everything seen on browser screens when websites, web applications, or mobile apps are opened up, is implemented by front End developers. Responsiveness and performance are two main objectives of the Front End. The developer must ensure that the site is responsive i.e. it appears correctly on devices of all sizes no part of the website should behave abnormally irrespective of the size of the screen.

**Front end Languages:** The front end portion is built by using some languages which are discussed below:

* **HTML:** HTML stands for Hypertext Markup Language. It is used to design the front-end portion of web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. The markup language is used to define the text documentation within the tag which defines the structure of web pages.
* **CSS:** Cascading Style Sheets fondly referred to as CSS is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page.
* **JavaScript:** JavaScript is a famous scripting language used to create magic on the sites to make the site interactive for the user. It is used to enhancing the functionality of a website to running cool games and web-based software.

There are many other languages through which one can do front-end development depending upon the framework for example *Flutter* user *Dart*, *React* uses *JavaScript* and *Django* uses *Python*, and much more.

**Pre-requisite**

Hands-on knowledge of JavaScript, HTML and CSS is essential before working on the concepts for making of webpages. Make sure that you have the browser or chrome installed and running before opening website.

**Chapter 2**

**Technologies Used**

**HTML**

HTML is the **language in which most websites are written**. HTML is used to create pages and make them functional.

The code used to make them visually appealing is known as CSS and we shall focus on this in a later tutorial. For now, we will focus on **teaching you how to build rather than design**.

HTML was first created by Tim Berners-Lee, Robert Cailliau, and others starting in **1989**. It stands for Hyper Text Markup Language.

Hypertext means that the document contains **links that allow the reader to jump to other places** in the document or to another document altogether. The latest version is known as HTML 5

A **Markup Language** is a way that computers speak to each other to control how text is processed and presented. To do this HTML uses two things: tags and **attributes**.

Tags and attributes are the basis of HTML.

They work together but perform different functions – it is worth investing 2 minutes in **differentiating the two**.

**Basic Construction of an HTML Page**

These tags should be placed underneath each other **at the top of every HTML page** that you create.

[<!DOCTYPE html>](https://html.com/tags/doctype/) — This tag **specifies the language** you will write on the page. In this case, the language is HTML 5.

[<html>](https://html.com/tags/html/) — This tag signals that from here on we are going to write in HTML code.

[<head>](https://html.com/tags/head/) — This is where all the **metadata for the page** goes — stuff mostly meant for search engines and other computer programs.

[<body>](https://html.com/tags/body/) — This is where the **content of the page** goes.

**CSS**

CSS (Cascading Style Sheets) is used to style and layout web pages — for example, to alter the font, color, size, and spacing of your content, split it into multiple columns, or add animations and other decorative features. This module provides a gentle beginning to your path towards CSS mastery with the basics of how it works, what the syntax looks like, and how you can start using it to add styling to HTML.

[**CSS building blocks**](https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks)

This module carries on where [CSS first steps](https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps) left off — now you've gained familiarity with the language and its syntax, and got some basic experience with using it, it's time to dive a bit deeper. This module looks at the cascade and inheritance, all the selector types we have available, units, sizing, styling backgrounds and borders, debugging, and lots more.

The aim here is to provide you with a toolkit for writing competent CSS and help you understand all the essential theory, before moving on to more specific disciplines like [text styling](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text) and [CSS layout](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout).

[**Styling text**](https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text)

With the basics of the CSS language covered, the next CSS topic for you to concentrate on is styling text — one of the most common things you'll do with CSS. Here we look at text styling fundamentals, including setting font, boldness, italics, line and letter spacing, drop shadows, and other text features. We round off the module by looking at applying custom fonts to your page, and styling lists and links.

[**CSS layout**](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout)

At this point we've already looked at CSS fundamentals, how to style text, and how to style and manipulate the boxes that your content sits inside. Now it's time to look at how to place your boxes in the right place in relation to the viewport, and to each other. We have covered the necessary prerequisites so we can now dive deep into CSS layout, looking at different display settings, modern layout tools like flexbox, CSS grid, and positioning, and some of the legacy techniques you might still want to know about.

[**Use CSS to solve common problems**](https://developer.mozilla.org/en-US/docs/Learn/CSS/Howto)

This module provides links to sections of content explaining how to use CSS to solve common problems when creating a web page.

**Cascading Style Sheets** (**CSS**) is a [style sheet language](https://en.wikipedia.org/wiki/Style_sheet_language) used for describing the [presentation](https://en.wikipedia.org/wiki/Presentation_semantics) of a document written in a [markup language](https://en.wikipedia.org/wiki/Markup_language) such as [HTML](https://en.wikipedia.org/wiki/HTML).[[1]](https://en.wikipedia.org/wiki/CSS#cite_note-1) CSS is a cornerstone technology of the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web), alongside HTML and [JavaScript](https://en.wikipedia.org/wiki/JavaScript).[[2]](https://en.wikipedia.org/wiki/CSS#cite_note-2)

CSS is designed to enable the separation of presentation and content, including [layout](https://en.wikipedia.org/wiki/Page_layout), [colors](https://en.wikipedia.org/wiki/Color), and [fonts](https://en.wikipedia.org/wiki/Typeface).[[3]](https://en.wikipedia.org/wiki/CSS#cite_note-3) This separation can improve content [accessibility](https://en.wikipedia.org/wiki/Accessibility), provide more flexibility and control in the specification of presentation characteristics, enable multiple [web pages](https://en.wikipedia.org/wiki/Web_page) to share formatting by specifying the relevant CSS in a separate .css file which reduces complexity and repetition in the structural content as well as enabling the .css file to be [cached](https://en.wikipedia.org/wiki/Cache_(computing)) to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or [screen reader](https://en.wikipedia.org/wiki/Screen_reader)), and on [Braille-based](https://en.wikipedia.org/wiki/Braille_display) tactile devices. CSS also has rules for alternate formatting if the content is accessed on a [mobile device](https://en.wikipedia.org/wiki/Mobile_device).[[4]](https://en.wikipedia.org/wiki/CSS#cite_note-4)

The name *cascading* comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

**JavaScript**

**Javascript** is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning Javascript:

* Javascript is the most popular **programming language** in the world and that makes it a programmer’s great choice. Once you learnt Javascript, it helps you developing great front-end as well as back-end softwares using different Javascript based frameworks like jQuery, Node.JS etc.
* Javascript is everywhere, it comes installed on every modern web browser and so to learn Javascript you really do not need any special environment setup. For example Chrome, Mozilla Firefox , Safari and every browser you know as of today, supports Javascript.
* Javascript helps you create really beautiful and crazy fast websites. You can develop your website with a console like look and feel and give your users the best Graphical User Experience.
* JavaScript usage has now extended to mobile app development, desktop app development, and game development. This opens many opportunities for you as Javascript Programmer.
* Due to high demand, there is tons of job growth and high pay for those who know JavaScript. You can navigate over to different job sites to see what having JavaScript skills looks like in the job market.
* Great thing about Javascript is that you will find tons of frameworks and Libraries already developed which can be used directly in your software development to reduce your time to market.

There could be 1000s of good reasons to learn Javascript Programming. But one thing for sure, to learn any **programming language**, not only Javascript, you just need to code, and code and finally code until you become expert.

**BOOTSTRAP**

* Bootstrap is the most popular HTML, CSS and JavaScript framework for developing a responsive and mobile friendly website.
* It is absolutely free to download and use.
* It is a front-end framework used for easier and faster web development.
* It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many others.
* It can also use JavaScript plug-ins.
* It facilitates you to create responsive designs.
* Bootstrap was developed by Mark Otto and Jacob Thornton at Twitter.It was released as an open source product in August 2011 on GitHub.
* In June 2014 Bootstrap was the No.1 project on GitHub.

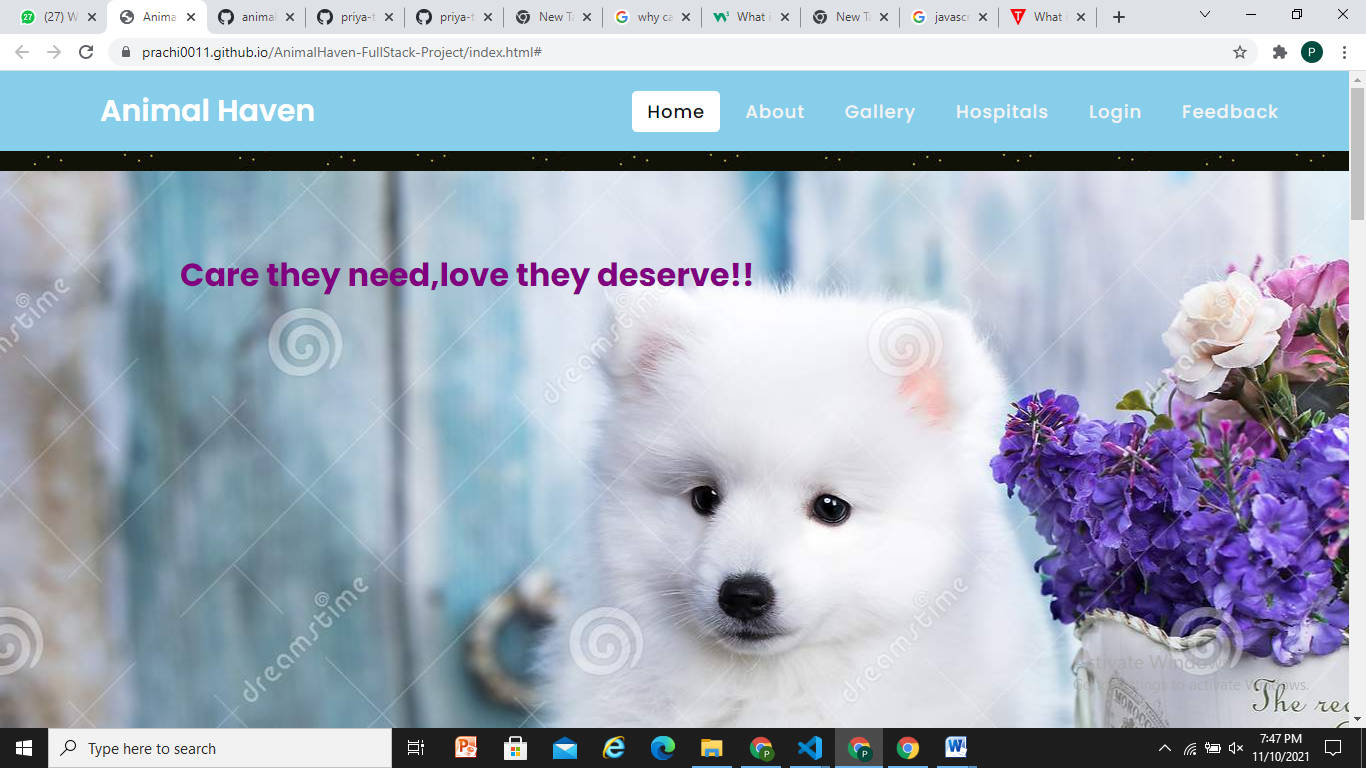
### What is a responsive website

A website is called responsive website which can automatically adjust itself to look good on all devices, from smart phones to desktops etc.

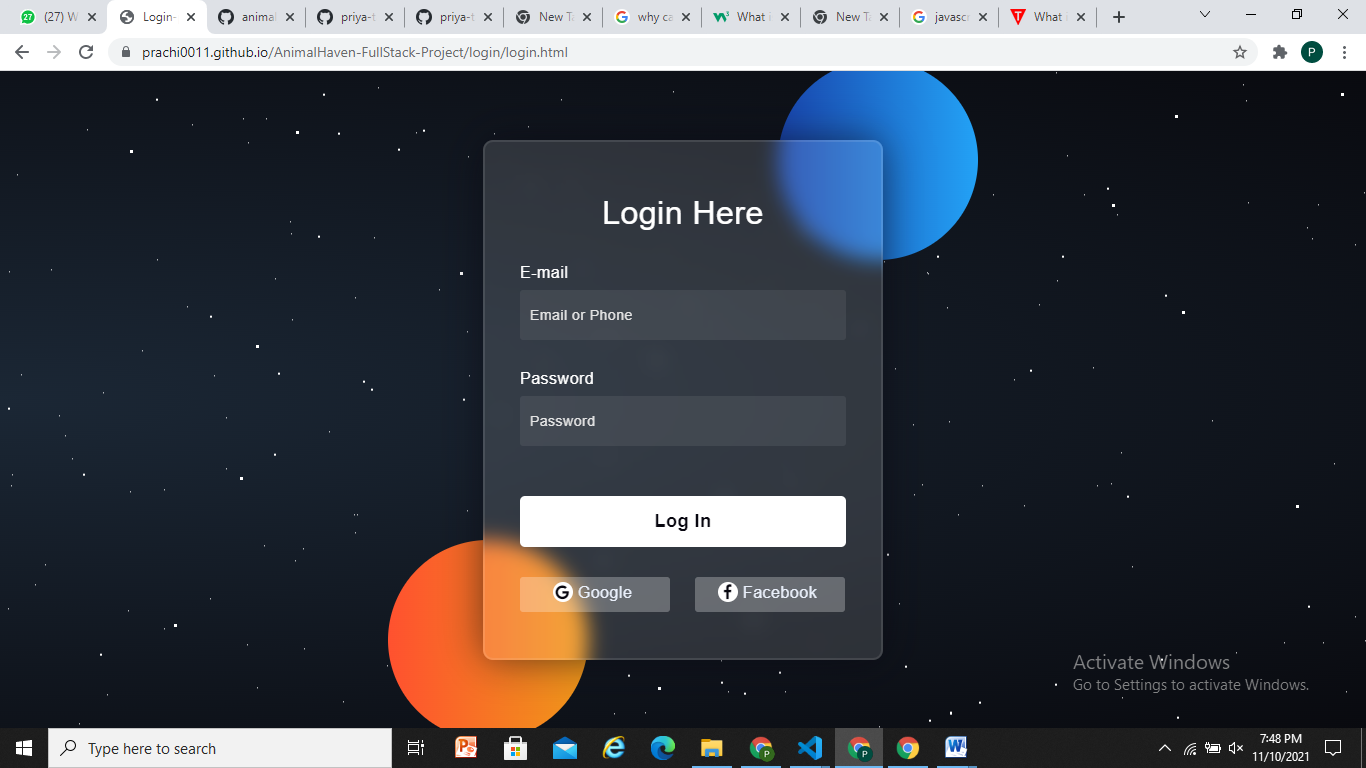
**Chapter 3**

**List of Figures**

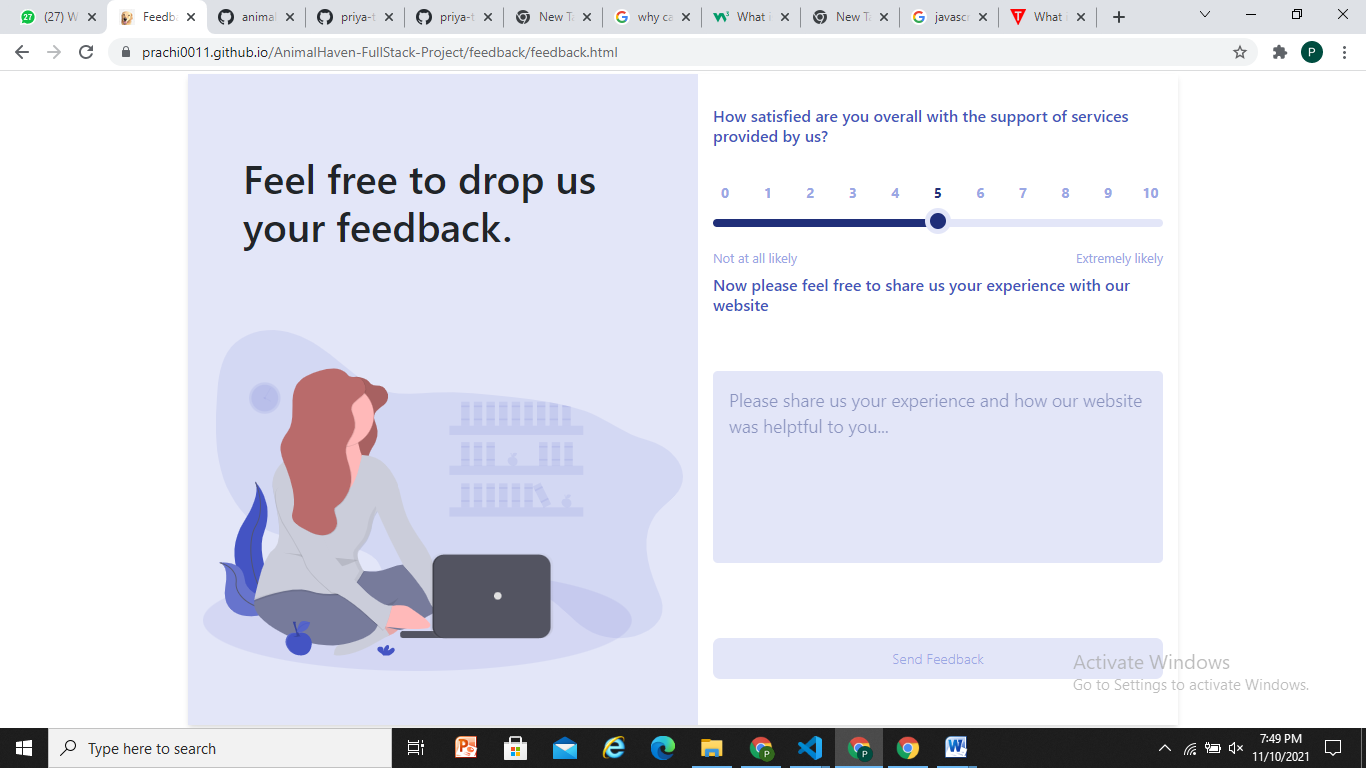
1. **HOME PAGE**



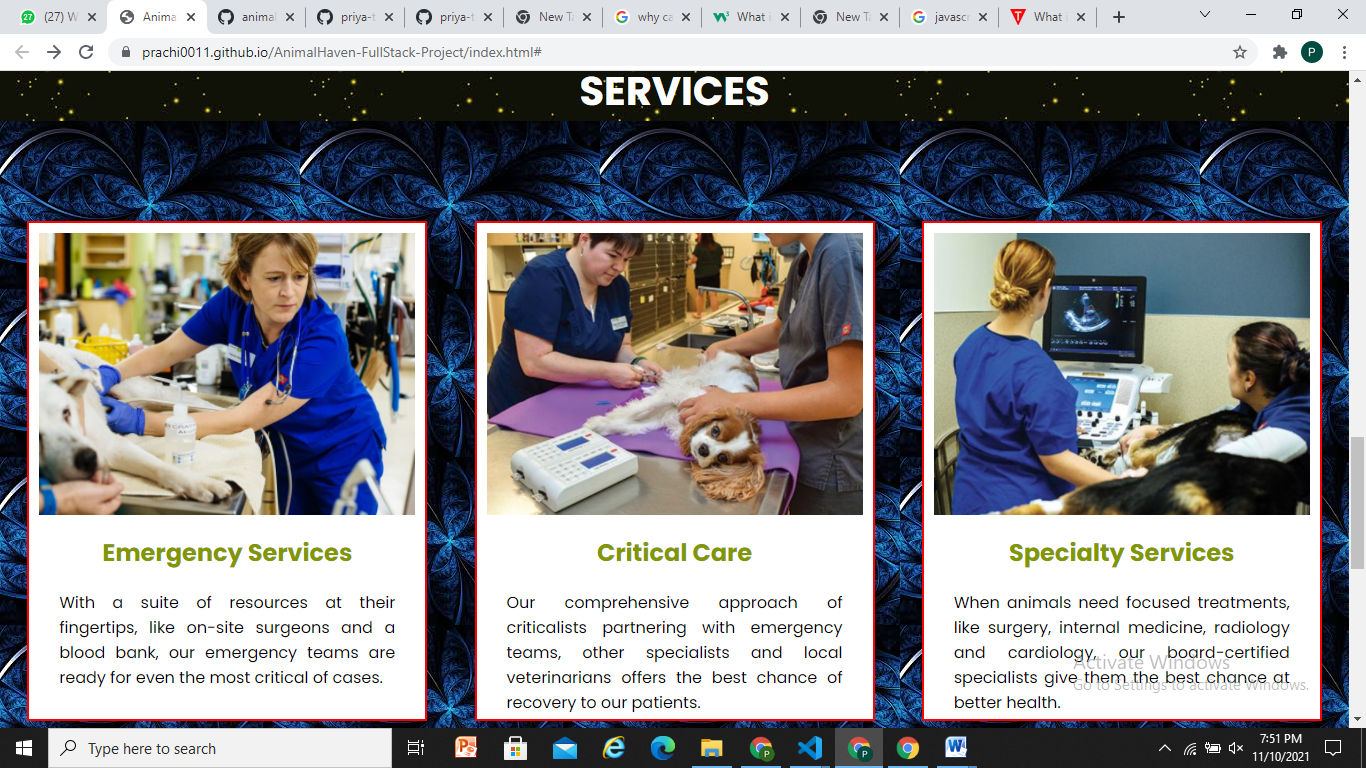
1. **Login Page**



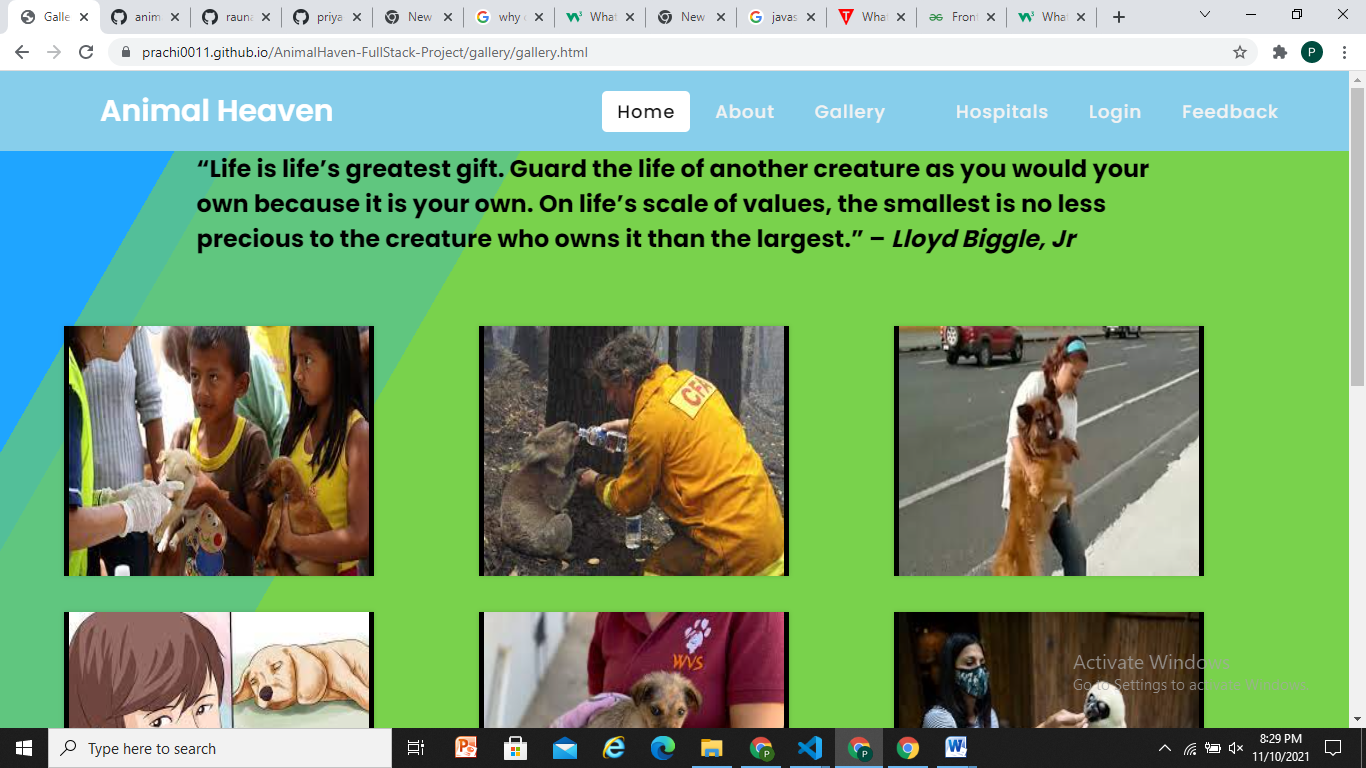
**3.Feedback and suggestions**



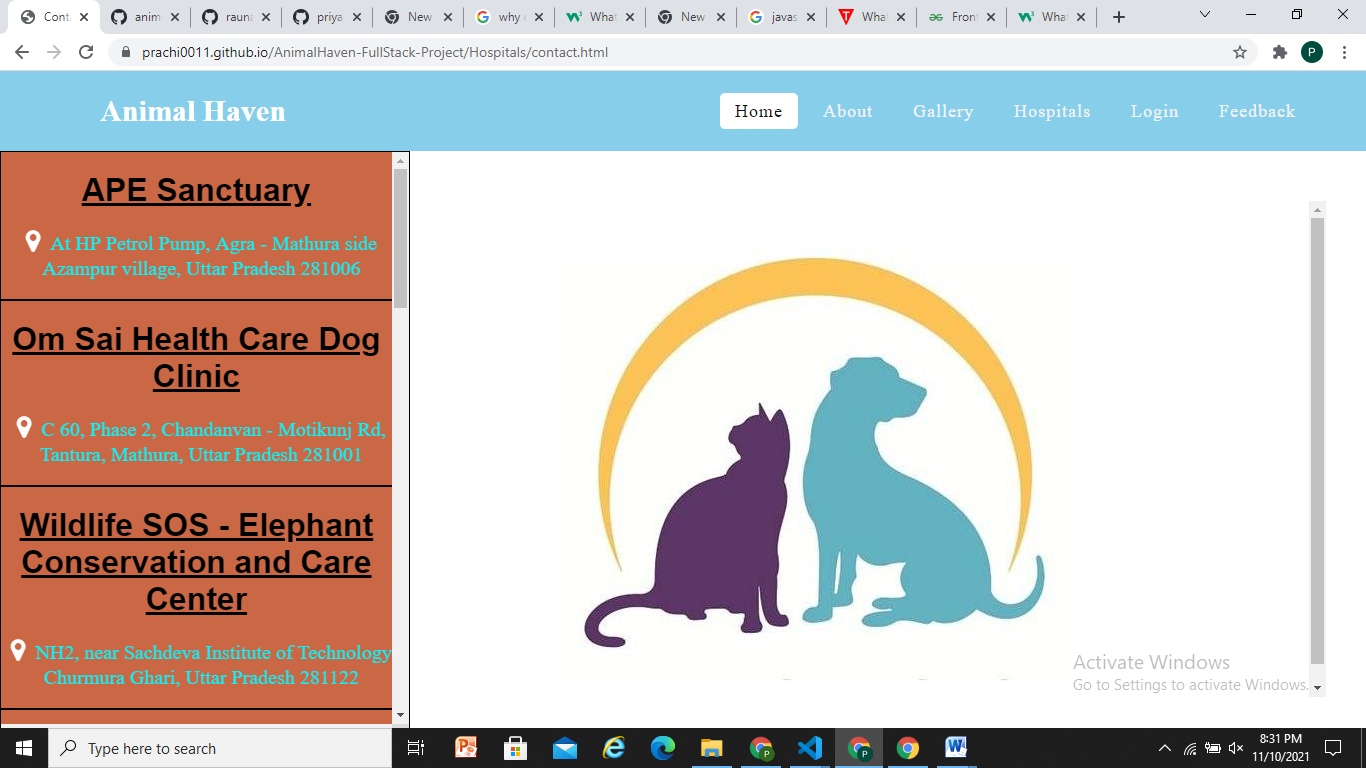
1. **Services**



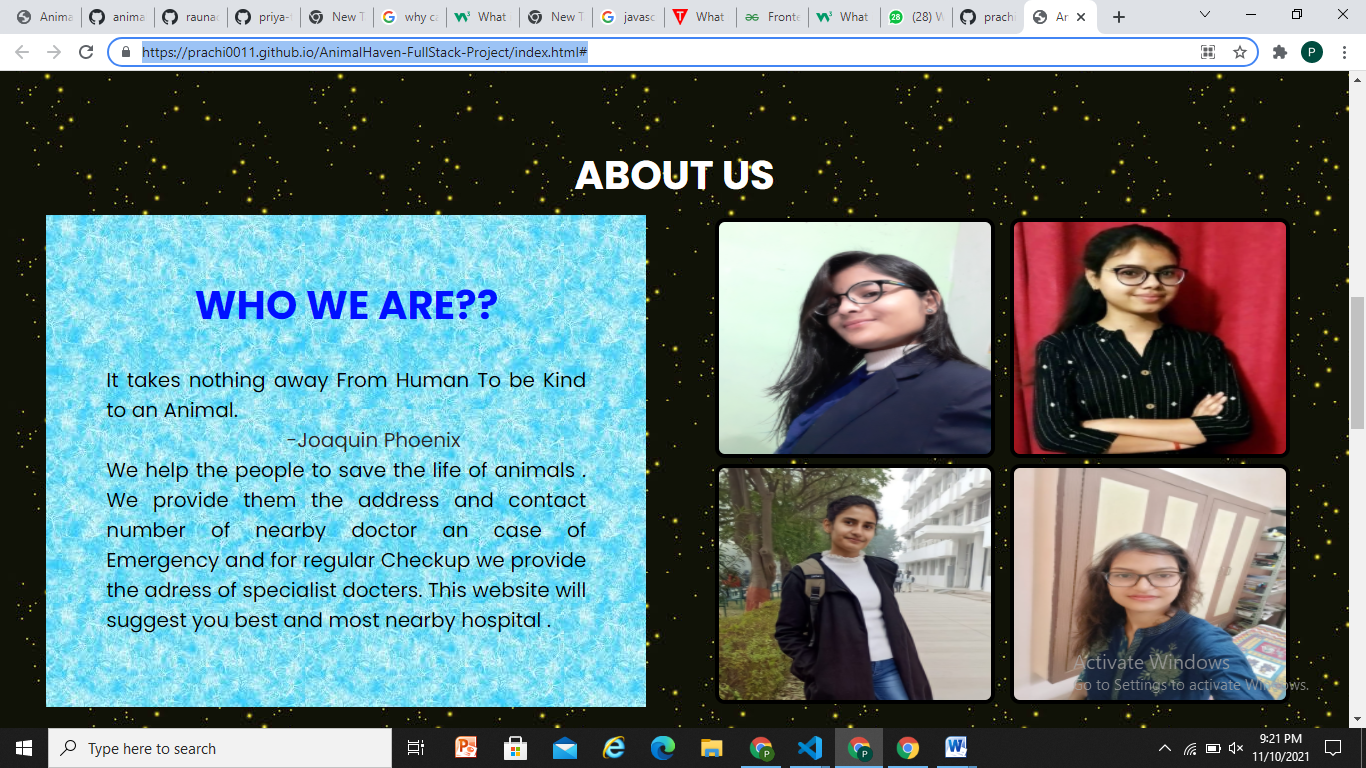
**5: Gallary**



**6. Hospitals**



**7 : About Us**



**Chapter 4**

**Software Testing**

Once source code has been generated, software must be tested to uncover as many errors as possible before delivery. It is very important to work the system successfully and achieve high quality of software. Testing include designing a series of test cases that have a high likelihood of finding errors by applying software-testing techniques. System testing makes logical assumptions that if all the parts of the system are correct, the goal will be successfully achieved. The system should be checked logically. Validations and cross checks should be there. Avoid duplications of record that cause redundancy of data. In other Words, Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. It is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

The preliminary goal of implementation is to write source code and internal documentation so that conformance of the code to its specifications can be easily verified, and so that debugging, testing and modifications are eased. This goal can be achieved by making the source code as clear and straightforward as possible. Simplicity, clarity and elegance are the hallmark of good programs, obscurity, cleverness, and complexity are indications of inadequate design and misdirected thinking. Source code clarity is enhanced by structured coding techniques, by good coding style, by, appropriate supporting documents, by good internal comments, and by feature provided in modern programming languages. The implementation team should be provided with a well-defined set of software requirement, an architectural design specification, and a detailed design description. Each team member must understand the objectives of implementation.

4.1 TERMINOLOGY

Error The term error is used in two ways. It refers to the difference between the actual output of software and the correct output, in this interpretation, error is essential a measure of the difference between actual and ideal. Error is also to used to refer to human action that result in software containing a defect or fault.

Fault is a condition that causes to fail in performing its required function. A fault is a basic reason for software malfunction and is synonymous with the commonly used term Bug.

Failure is the inability of a system or component to perform a required function according to its specifications. A software failure occurs if the behavior of the software is the different from the specified behavior. Failure may be caused due to functional or performance reasons.

4.2 TYPES OF TESTING

**a. Unit Testing** The term unit testing comprises the sets of tests performed by an individual programmer prior to integration of the unit into a larger system. A program unit is usually small enough that the programmer who developed it can test it in great detail, and certainly in greater detail than will be possible when the unit is integrated into an evolving software product. In the unit testing the programs are tested separately, independent of each other. Since the check is done at the program level, it is also called program teasing.

**b. Module Testing** A module and encapsulates related component. So can be tested without other system module.

**c. Subsystem Testing** Subsystem testing may be independently design and implemented common problems are sub-system interface mistake in this checking we concenton it. There are four categories of tests that a programmer will typically perform on a program unit.

i Functional test

ii Performance test

iii Stress test

iv Structure test

**Functional Test** Functional test cases involve exercising the code with Nominal input values for which expected results are known; as well as boundary values (minimum values, maximum values and values on and just outside the functional boundaries) and special values.

**Performance Test** Performance testing determines the amount of execution time spent in various parts of the unit, program throughput, response time, and device utilization by the program unit. A certain amount of avoid expending too much effort on fine-tuning of a program unit that contributes little to the overall performance of the entire system. Performance testing is most productive at the subsystem and system levels.

**Stress Test** Stress test are those designed to intentionally break the unit. A great deal can be learned about the strengths and limitations of a program by examining the manner in which a program unit breaks.

**Structure Test** Structure tests are concerned with exercising the internal logic of a program and traversing particular execution paths. Some authors refer collectively to functional performance and stress testing as “black box” testing. While structure testing is referred to as “white box” or “glass box” testing. The major activities in structural testing are deciding which path to exercise, deriving test date to exercise those paths, determining the test coverage criterion to be used, executing the test, and measuring the test coverage achieved when the test cases are exercised.

**Chapter 5**

**Conclusion**

We have completed our project within time limit with the coordination of our team members under the supervision of our mentor Mr. Pankaj Kapoor.

live page link

[https://prachi0011.github.io/AnimalHaven-FullStack-Project/index.html#](https://prachi0011.github.io/AnimalHaven-FullStack-Project/index.html)

Our project repository is available at

[**https://github.com/prachi0011/AnimalHaven-FullStack-Project**](https://github.com/prachi0011/AnimalHaven-FullStack-Project)

**Chapter 6**

**Bibliography**

[**www.google.com**](http://www.google.com/)

[**www.geeksforgeeks.org**](http://www.geeksforgeeks.org/)

[**www.youtube.com**](http://www.youtube.com/)

[**www.w3schools.com**](http://www.w3schools.com/)

[**www.beta-labs.in**](http://www.beta-labs.in/)