A PROJECT REPORT

On

STRAY CARE

Submitted in partial fulfilment of the requirement of
University of Mumbai for

Internet Programming Mini Project
In
Information Technology

Submitted By
Aditya Chattikal
Akash Khatkale
Rohit Chaudhari
Kiran Kumawat

Department Of Information Technology

PILLAI COLLEGE OF ENGINEERING

New Panvel – 410 206

UNIVERSITY OF MUMBAI

Academic Year 2020 – 21

DEPARTMENT OF INFORMATION TECHNOLOGY
Pillai College of Engineering
New Panvel – 410 206

CERTIFICATE

This is to certify that the requirements for the report entitled 'Stray Care: A platform to protect stray animals ' have been successfully completed by the following students:

Name	**Roll No.**
Aditya Chattikal	06
Akash Khatkale	18
Rohit Chaudhari	08
Kiran Kumawat	20

in partial fulfillment of Internet Programming mini Project in the Department of Information Technology, Pillai College of Engineering, New Panvel – 410 206 during the Academic Year 2019 – 2020.

Prof. Dhiraj Amin

DEPARTMENT OF INFORMATION TECHNOLOGY
Pillai College of Engineering
New Panvel – 410 206

PROJECT APPROVAL FOR

This project entitled "Stray Care" by Aditya, Akash, Rohit, and Kiran are approved for the degree of Bachelor of Engineering in Information Technology.

Examiners:

1. ________________

2. ________________

3. ________________

4. ________________

5. ________________

Date:

DEPARTMENT OF INFORMATION TECHNOLOGY

Pillai College of Engineering

New Panvel – 410 206

DECLARATION

We declare that this written submission for Internet Programming Mini Project entitled "Stray Care" represent our ideas in our own words and where others' ideas or words have been included. We have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any ideas / data / fact / source in our submission. We understand that any violation of the above will cause for disciplinary action by institute and also evoke penal action from the sources which have not been properly cited or from whom prior permission have not been taken when needed.

Project Group Members:

[illegible] Akash Khatkale

[illegible] Aditya Chattikal

kiran Kiran Kumawat

[illegible] Rohit Chaudhari

Table of Contents

Abstract		i
1.	**Introduction**	
2.	**Requirement Analysis**	
3.	**Wireframe**	
4	**Using different types of CSS**	
5	**Responsive design using media queries**	
6	**Adding parallax effect in web page**	
7	**Embedding Maps and Video**	
8	**HTML5 based form validation**	
9	**JavaScript based form validation**	
10	**Server side programming using PHP**	
11	**PHP form validation**	
12	**PHP MySQL database operations**	
13	**RIA using AJAX**	
14	**Web hosting**	
	Acknowledgement	

Abstract

Stray Care will help all stray animals to get all sorts of medical help in case of an accident/ injury, in this app a user can upload a picture of any injured dog or animal near his / her vicinity and as soon as the picture of the injured animals is uploaded it will sent to the nearest VET doctors and Municipal corporation for immediate help, customers can later on track the status of the animals they helped to if needed. The main intention of the app is to provide help and medical support to as many animals as possible

Chapter 1
Introduction

Nowadays there are many stray animals which are injured and don't have proper treatment , As most of the people have no idea about the animal doctors or their hospitals but there needs to be development in everything so we have created an app which just helps you to get the easiest solution for this problem.

Our app named stray care provides you a facility in which you just upload the image of the injured animal and a nearby doctor will contact that person who has uploaded the image. The user needs to sign up for the same which does not ask for hectic details, just your name, email and password and you can upload the image whenever you want. By this way there will be some awareness in the society . Once you upload the image the app just uploads your location and the doctors nearing that location and using the same app gets the feed on their app by which they can take the appointment and contact the person. In this way this app helps to cure the injured stray animal.

Chapter 2

Requirement Analysis

Requirement analysis techniques play very important role in any of your web or mobile app development project's success. Are you properly analyzing the requirements before you/your team start working on the web app?

Most mid to large scale projects fail or delayed? It's because of the faulty or improper requirement analysis techniques for the web app.

1. Functional requirements
2. Non-functional requirements

Functional requirements

This might be the heart of requirement analysis techniques. You might want to thoroughly understand the functional requirements your client has for the project even before sending the price quote. How would you quote for something before even knowing what functionality is required and how much time and effort you may need to address them.

How to analyze?

The beginners fall a short with functional requirement gathering. They generally try to collect all the information at once and look for the technical ways to address those needs. Don't you think it's quite early to start working on the project after having a brief requirement overview? How much information(in depth) do you think you can collect in few(2 or 3) meetings?

Rather focus on collecting the core idea or functionality the client is looking. If you dive a little deeper you will see there is a problem which he/she wants to solve with the project. Ideally, we would understand the problem first.

After getting a good idea of the problem, We would focus on the very core functionality that client is proposing to solve the problem. No fancy features or things come into the picture just yet. Simply the problem and functionality that addresses the problem.

This should be the area where you focus the most. The success of your project development relies on how efficiently your solution is helping your client solving the core problem. Those extra UI effects, animations, reporting features, and other bells and whistles make no sense if your solution isn't solving the problem properly.

Non-functional requirements

Non-functional requirements are those which don't really affect the core problem you are solving with the application. But they are pretty important to your client. Here are some of the non-functional requirements we should give place into our requirement analysis techniques.

Legal Aspects:

If we are working on the eCommerce web application development, we would ask the client about the compliance certificate he/she might need. Who will get them and How they need to be integrated? If there are transactions attached to the web application, what about the return policy? Who will provide the privacy, cookie, and return policy?

Performance:

Sometimes, clients have performance concerns that the web app should load under X seconds. The app should work with the mobile device. The should be able to bare X number of concurrent requests and so on. All the performance related queries and doubts should be cleared in this phase.

Support:

What kind of support does the client need? How would you be able to provide the support? What will be included in the support and the fees for the support? It may sound small but it's one of the essential requirement analysis techniques to build better trust and relationship with your client.

Security:

What are the security requirements? What security standards needs to be followed? How opt the

SSL certification? What does other security compliance certificates the client need and who will provide them? Make the security an integral part of your requirement gathering techniques.

UI Preference:

If you are working on a client/customer facing web application then your client might have some design, color, and typography preference. Make sure you include them in your requirement gathering techniques as well. It will help you create the design next to his/her preference. Using an UI framework will help you save quite a good amount of time. Bootstrap is a trending responsive web design framework. Learn more about what is bootstrap and how to use it.

List of Functional Requirements:

1. User should be able to post a pic of stray animal
2. Doctor should be able to see all the nearby users' posts
3. Users should be able to login easily
4. Users should be able to view their post status in profile
5. Doctors should have an option to accept whether he can treat the stray animal or not.

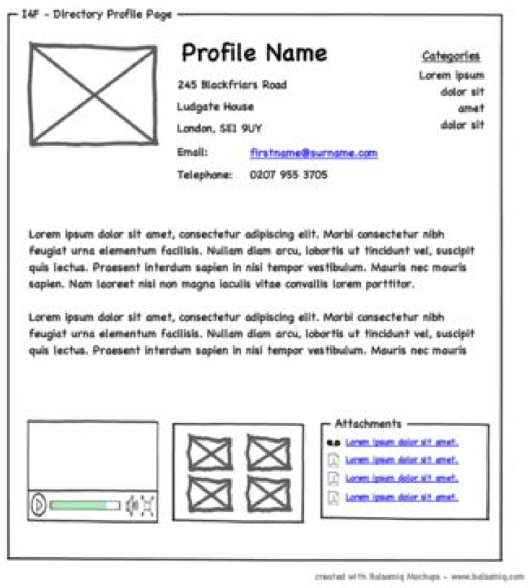
List of Non Functional Requirements:

1. Terms and conditions
2. About us
3. Privacy Policy
4. Support

Chapter 3

Wireframe

A website wireframe, also known as a page schematic or screen blueprint, is a visual guide that represents the skeletal framework of a website. Wireframes are created for the purpose of arranging elements to best accomplish a particular purpose. The purpose is usually being informed by a business objective and a creative idea. The wireframe depicts the page layout or arrangement of the website's content, including interface elements and navigational systems, and how they work together. The wireframe usually lacks typographic style, color, or graphics, since the main focus lies in functionality, behavior, and priority of content. In other words, it focuses on what a screen does, not what it looks like. Wireframes can be pencil drawings or sketches on a whiteboard, or they can be produced by means of a broad array of free or commercial software applications. Wireframes are generally created by business analysts, user experience designers, developers, visual designers, and by those with expertise in interaction design, information architecture and user research.


I4F - Directory Profile Page
Profile Name
245 Blackfriars Road
Ludgate House
London, SE1 9UY
Email: firstname@surname.com
Telephone: 0207 955 3705
Categories
Lorem ipsum
dolor sit
amet
dolor sit
Attachments
Lorem ipsum dolor sit amet.
Lorem ipsum dolor sit amet.
Lorem ipsum dolor sit amet.
Lorem ipsum dolor sit amet.
created with Balsamiq Mockups - www.balsamiq.com

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi consectetur nibh feugiat urna elementum facilisis. Nullam diam arcu, lobortis ut tincidunt vel, suscipit quis lectus. Praesent interdum sapien in nisi tempor vestibulum. Mauris nec mauris sapien. Nam laoreet nisi non magna iaculis vitae convallis lorem porttitor.

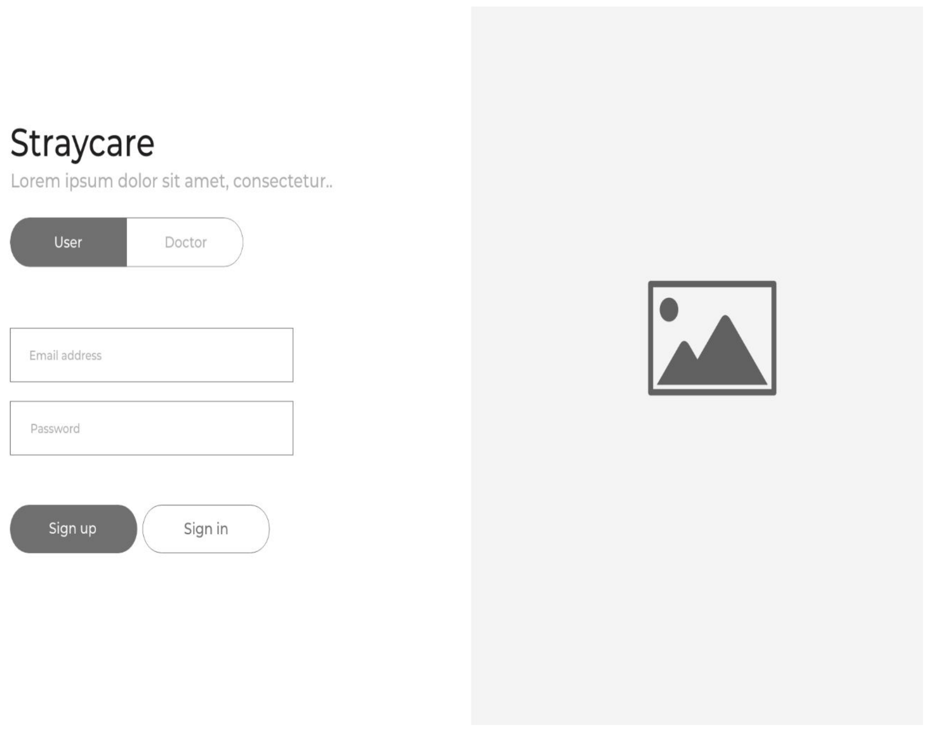
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi consectetur nibh feugiat urna elementum facilisis. Nullam diam arcu, lobortis ut tincidunt vel, suscipit quis lectus. Praesent interdum sapien in nisi tempor vestibulum. Mauris nec mauris

Wireframes focus on:

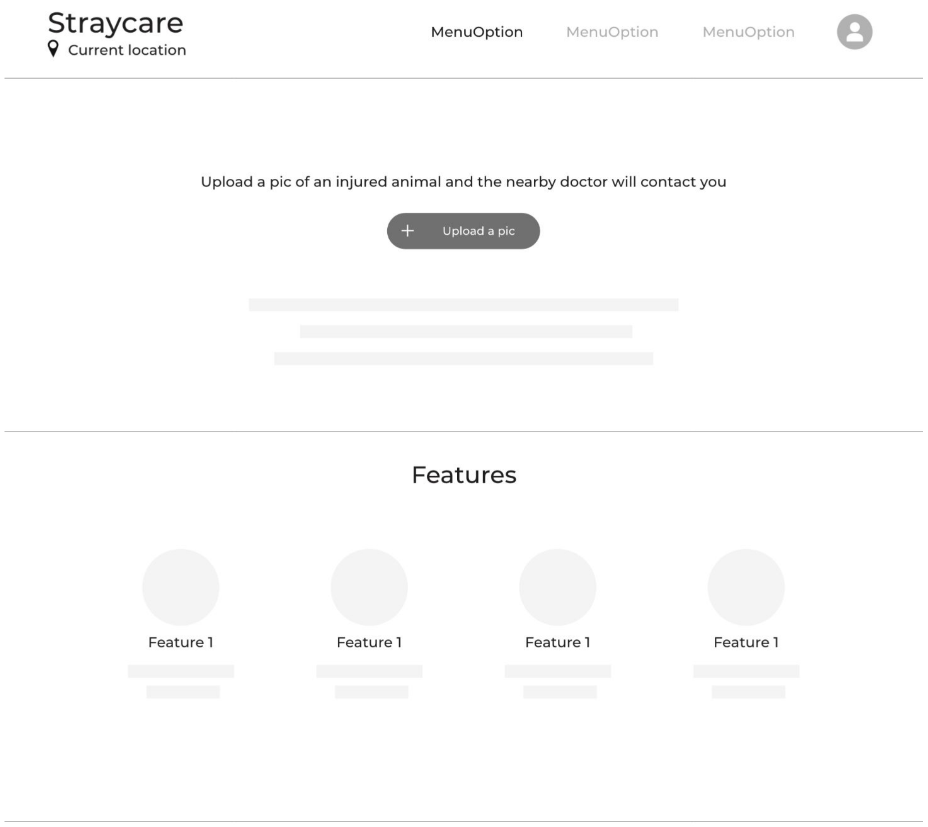
1. The range of functions available
2. The relative priorities of the information and functions

3. The rules for displaying certain kinds of information
4. The effect of different scenarios on the display

Log in page

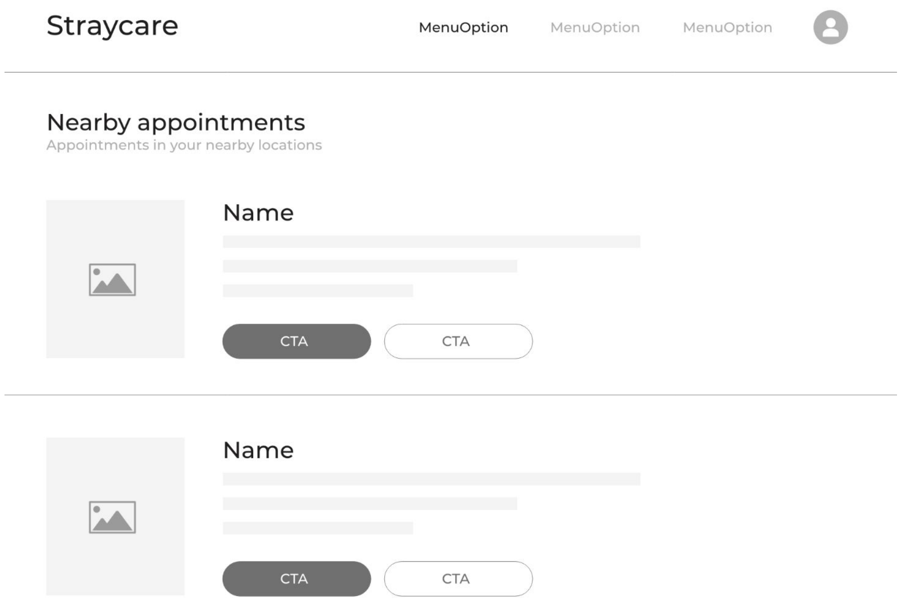


Home page



Upload a pic of an injured animal and the nearby doctor will contact you

Doctor page



Chapter 4

Using different types of CSS

Cascading Style Sheets (CSS) is a simple mechanism for adding style (e.g., fonts, colors, spacing) to Web documents. CSS is a language that describes the style of an HTML document. CSS describes how HTML elements should be displayed.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Separation of formatting and content also makes it possible 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), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

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. The CSS specifications are maintained by the World Wide Web Consortium (W3C).

Syntax of CSS

A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule is made of three parts :

Selector − A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table> etc.

Property - A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be color, border etc.

Value - Values are assigned to properties. For example, color property can have value either red or

#F1F1F1 etc.

selector { property: value } selector { property: value, property: value }

h1 {‘color’:’blue’}

Types of css selectors:

- The element Selector :h1 {color: red;}
- The id Selector / type selector :#para1 {text-align: center;color: red;}
- The Descendant Selectors : ul li {color: red;}
- The class Selector : .center {text-align: center;color: red;}
- The Attribute Selectors : input[type = "text"]{color: #000000; }
- The Child Selectors : body > p {color: #000000; }
- The Universal Selectors : * {color: red;}
- The Adjacent Sibling Selector : H2+P {color: red;}

There are three ways of inserting a style sheet:

1. External style sheet
2. Internal style sheet
3. Inline style

External style sheet

The <link> element can be used to include an external stylesheet file in your HTML document. An external style sheet is a separate text file with .css extension. You define all the Style rules within this text file and then you can include this file in any HTML document using <link> element.

Here is the generic syntax of including external CSS file −

```
<head>
  <link type = "text/css" href = "..." media = "..." />
</head>
```

Consider a simple style sheet file with a name mystyle.css having the following rules −

```
h1, h2, h3 {
  color: #36C;
  font-weight: normal;
  letter-spacing: .4em;
  margin-bottom: 1em;
  text-transform: lowercase;
}
```

Now you can include this file mystyle.css in any HTML document as follows −

```
<head>
  <link type = "text/css" href = "mystyle.css" media = " all" />
</head>
```

Internal style sheet

You can put your CSS rules into an HTML document using the <style> element. This tag is placed inside <head>...</head> tags. Rules defined using this syntax will be applied to all the elements available in the document

<style> Attribute type =”text/css”   <style type=”text/css” > </style>

Specifies the style sheet language as a content-type (MIME type). This is required attribute.

media attribute <style type = "text/css" media = "all">

```
<head>
<style type=”text/css” >
body {
        background-color: linen;
}
h1 {
        color: maroon;
        margin-left: 40px;
}
</style>
```

</head>

Inline style

An inline style may be used to apply a unique style for a single element. To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

Syntax:

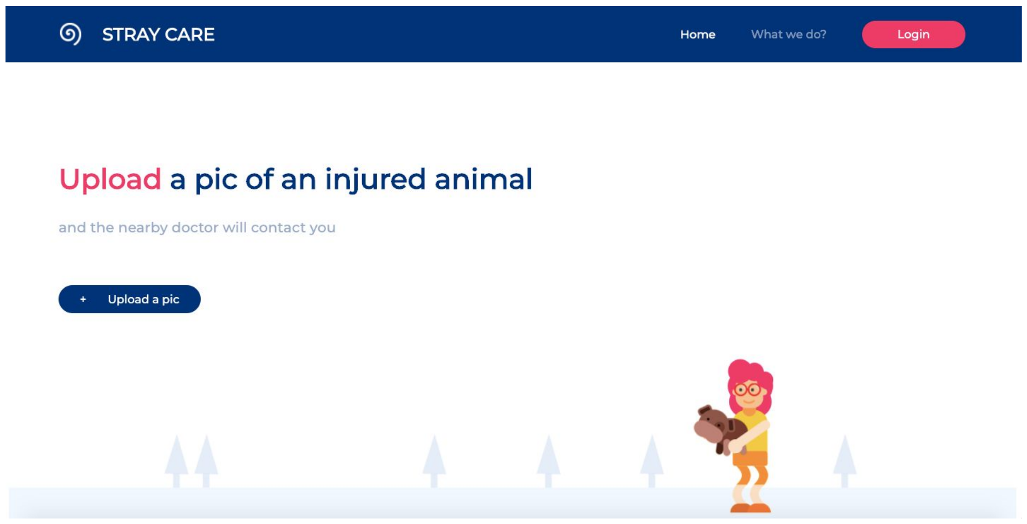
<element style = "...style rules....">

Attributes style "style rules"

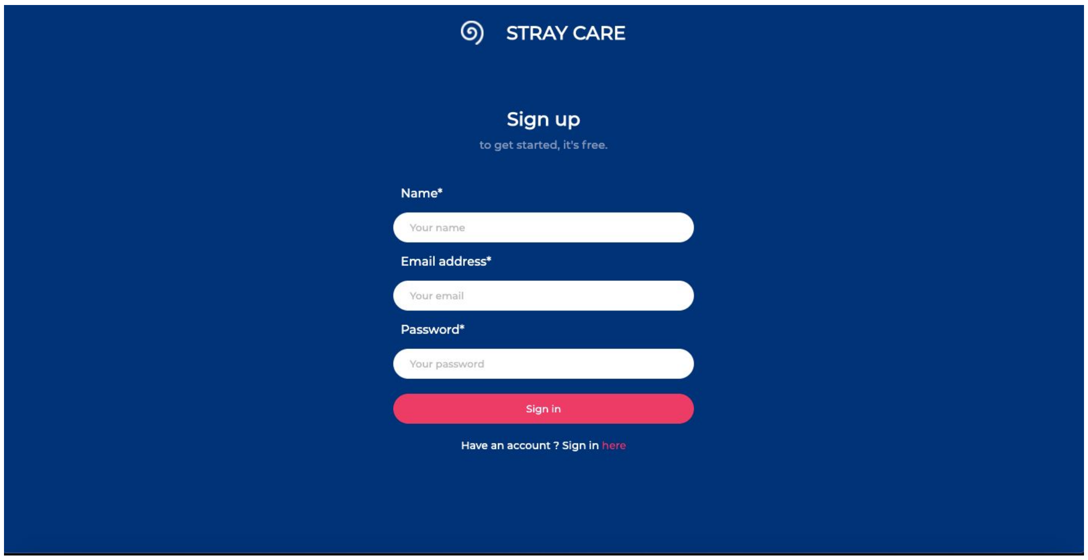
The value of style attribute is a combination of style declarations separated by semicolon (;)

<h1 style = "color:#36C;">

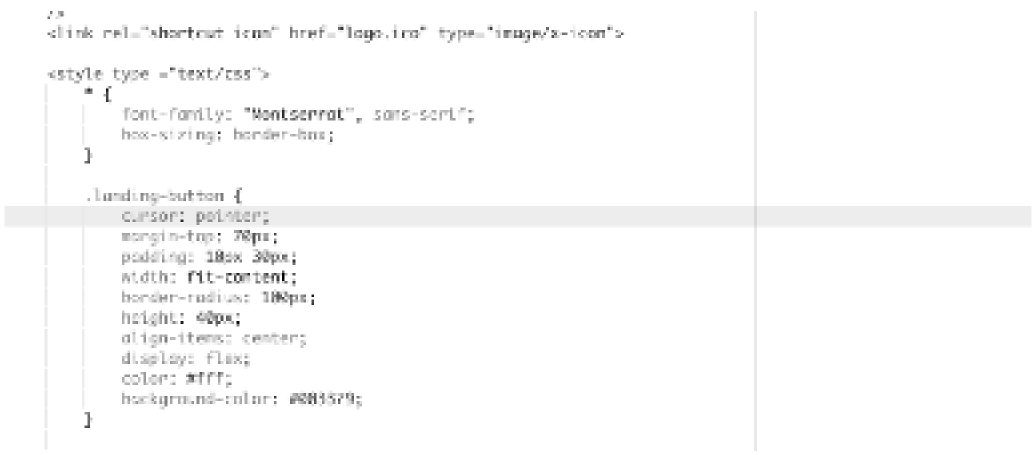
Home page :


STRAY CARE
Home
What we do?
Login
Upload a pic of an injured animal
and the nearby doctor will contact you
+ Upload a pic

Login page :

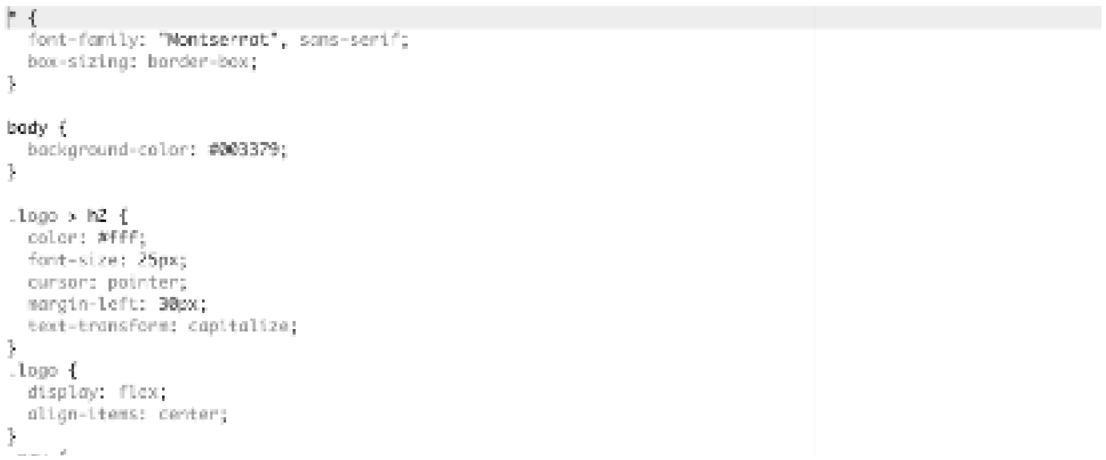


Internal css :



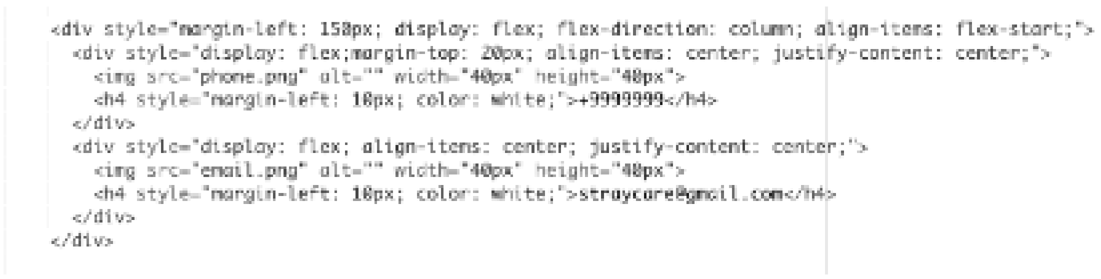
```
/*
<link rel="shortcut icon" href="logo.ico" type="image/x-icon">

<style type ="text/css">
    * {
        font-family: "Montserrat", sans-serif;
        box-sizing: border-box;
    }

    .landing-button {
        cursor: pointer;
        margin-top: 70px;
        padding: 10px 30px;
        width: fit-content;
        border-radius: 100px;
        height: 40px;
        align-items: center;
        display: flex;
        color: #fff;
        background-color: #003379;
    }
```

External css:



```
* {
  font-family: "Montserrat", sans-serif;
  box-sizing: border-box;
}

body {
  background-color: #003379;
}

.logo > h2 {
  color: #fff;
  font-size: 25px;
  cursor: pointer;
  margin-left: 30px;
  text-transform: capitalize;
}
.logo {
  display: flex;
  align-items: center;
}
```

Inline css:



```
<div style="margin-left: 150px; display: flex; flex-direction: column; align-items: flex-start;">
  <div style="display: flex;margin-top: 20px; align-items: center; justify-content: center;">
    <img src="phone.png" alt="" width="40px" height="40px">
    <h4 style="margin-left: 10px; color: white;">+9999999</h4>
  </div>
  <div style="display: flex; align-items: center; justify-content: center;">
    <img src="email.png" alt="" width="40px" height="40px">
    <h4 style="margin-left: 10px; color: white;">straycare@gmail.com</h4>
  </div>
</div>
```

Chapter 5
Responsive design using media queries

Media queries in CSS3 look at the capability of the device. Media queries can be used to check many things, such as:

- width and height of the viewport
- width and height of the device
- orientation (is the tablet/phone in landscape or portrait mode?)
- resolution

Using media queries is a popular technique for delivering a tailored style sheet to desktops, laptops, tablets, and mobile phones (such as iPhone and Android phones). Media queries are used for creating responsive web sites . You can also use media queries to specify that certain styles are only for printed documents or for screen readers (mediatype: print, screen, or speech).

In addition to media types, there are also media features. Media features provide more specific details to media queries, by allowing to test for a specific feature of the user agent or display device. For example, you can apply styles to only those screens that are greater, or smaller, than a certain width.

A media query consists of a media type and can contain one or more expressions, which resolve to either true or false.

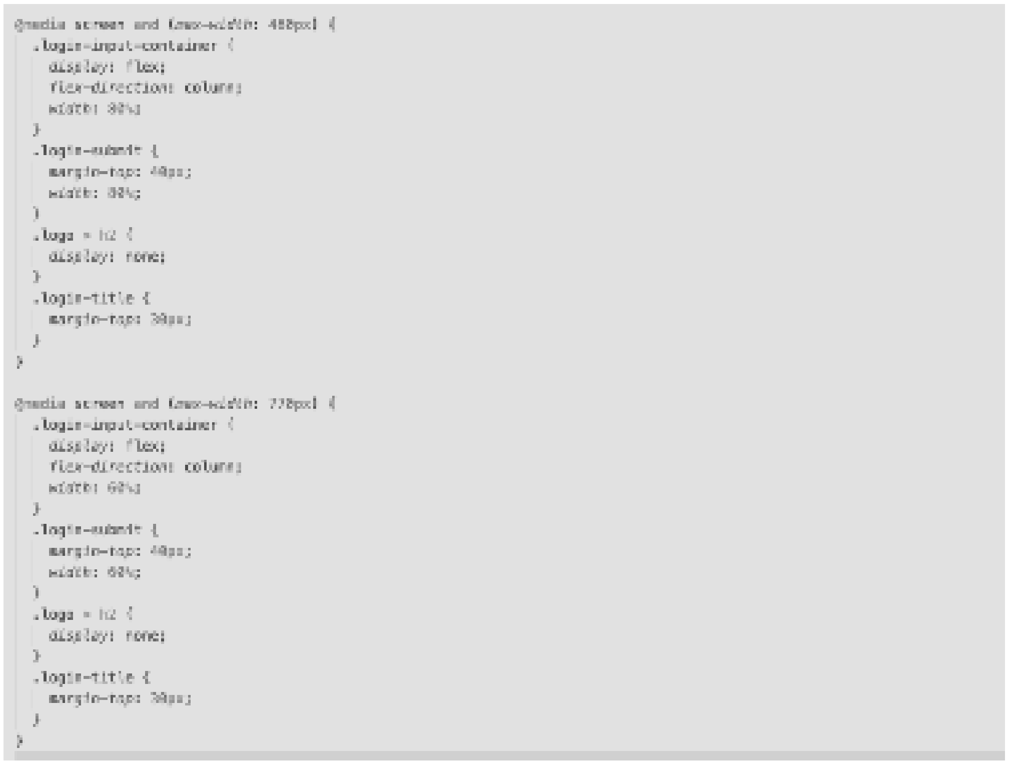
```
@media not|only mediatype and (expressions) {

        CSS-Code;

}
```

The result of the query is true if the specified media type matches the type of device the document is being displayed on and all expressions in the media query are true. When a media query is true, the corresponding style sheet or style rules are applied, following the normal cascading rules.

Unless you use the not or only operators, the media type is optional and the all type will be implied The following example changes the background-color to light green if the viewport is 480 pixels wide or wider (if the viewport is less than 480 pixels, the background-color will be pink):

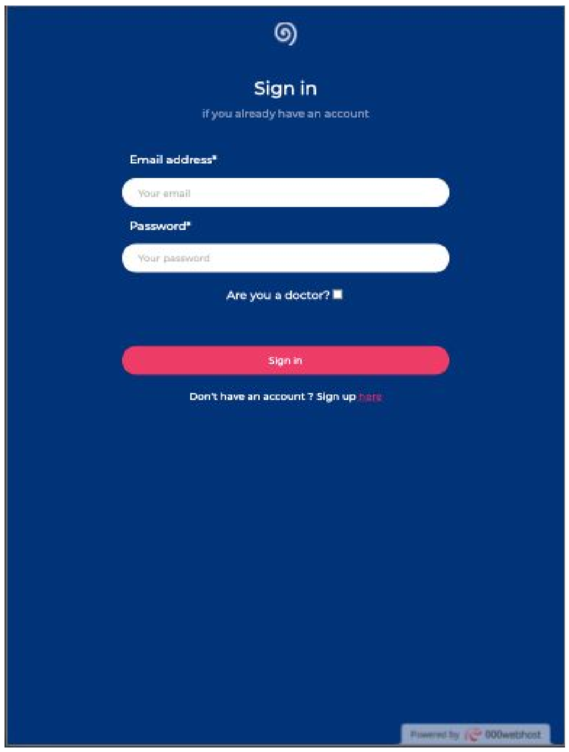
```
@media screen and (min-width: 480px) {

        body {

        background-color: lightgreen;

   }

}
```

Code(login.php):

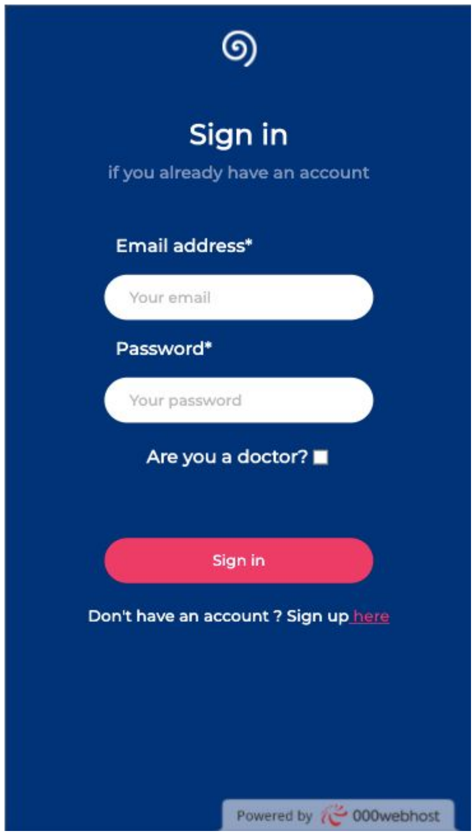


```
@media screen and (max-width: 480px) {
  .login-input-container {
    display: flex;
    flex-direction: column;
    width: 80%;
  }
  .login-submit {
    margin-top: 40px;
    width: 80%;
  }
  .logo > h2 {
    display: none;
  }
  .login-title {
    margin-top: 30px;
  }
}

@media screen and (max-width: 770px) {
  .login-input-container {
    display: flex;
    flex-direction: column;
    width: 60%;
  }
  .login-submit {
    margin-top: 40px;
    width: 60%;
  }
  .logo > h2 {
    display: none;
  }
  .login-title {
    margin-top: 30px;
  }
}
```

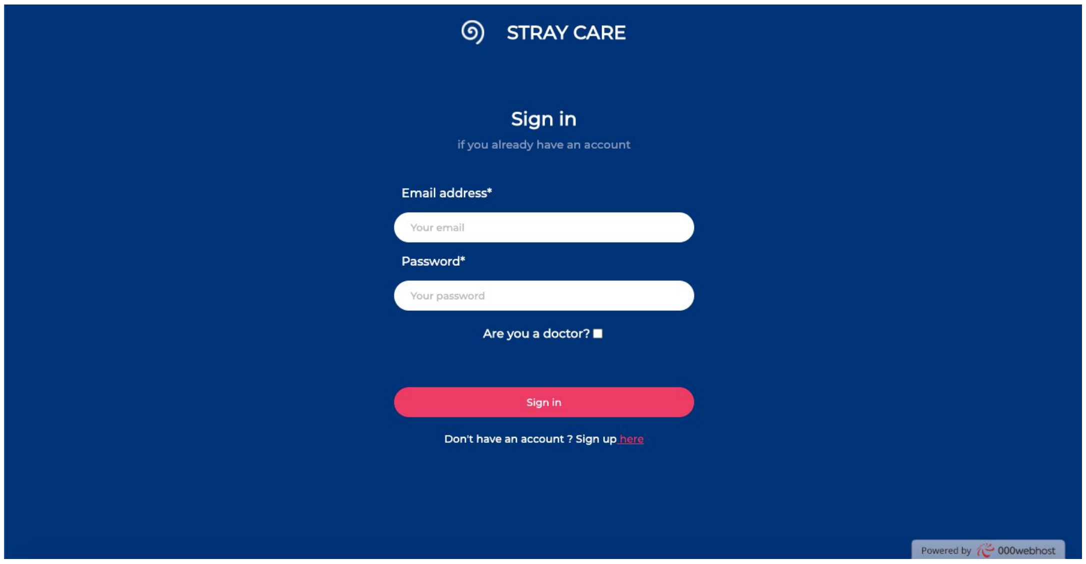
Tablet view(login.php):



Mobile view(login.php):



Laptop view(login.php):


STRAY CARE

Sign in

if you already have an account

Email address*

Your email

Password*

Your password

Are you a doctor? ☐

Sign in

Don't have an account ? Sign up here

Powered by 000webhost

Chapter 6

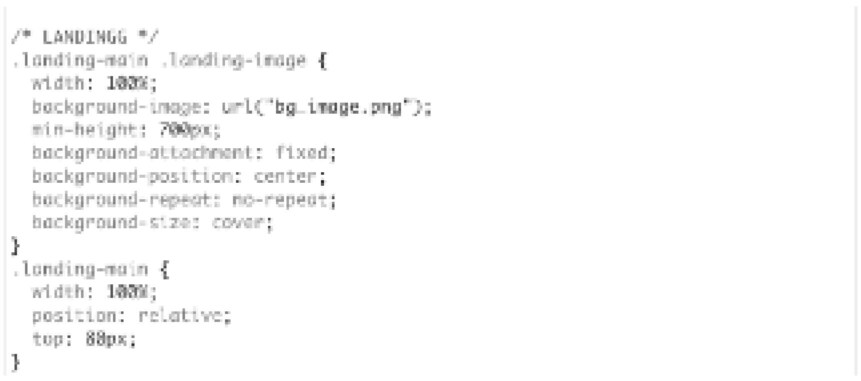
Adding parallax effect in web page

Parallax scrolling is a web site trend where the background content (i.e. an image) is moved at a different speed than the foreground content while scrolling. Parallax is an effect where the background content or image in this case, is moved at a different speed than the foreground content while scrolling.

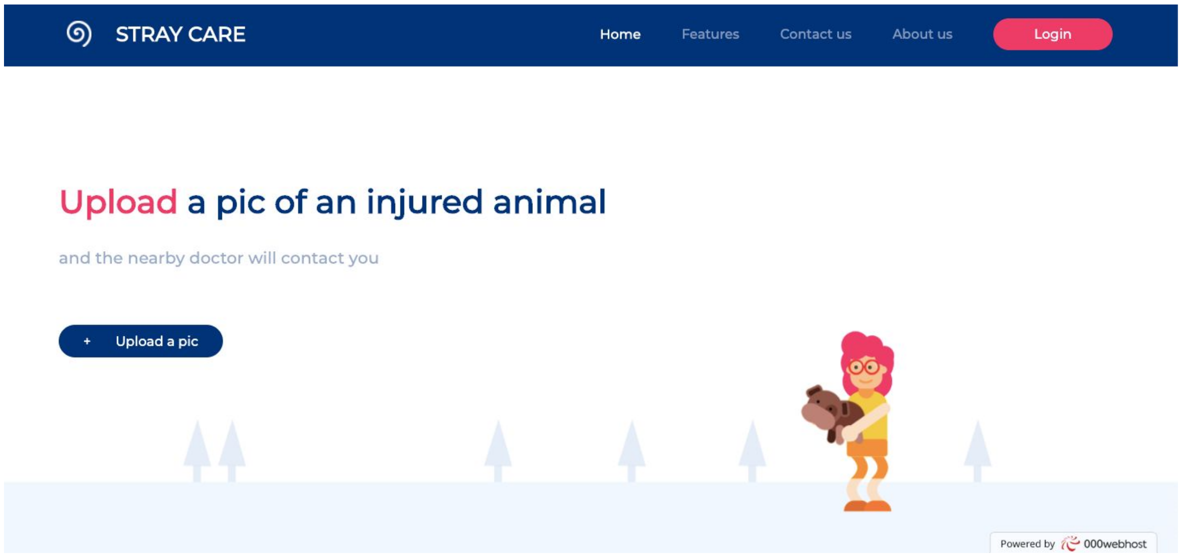
Example

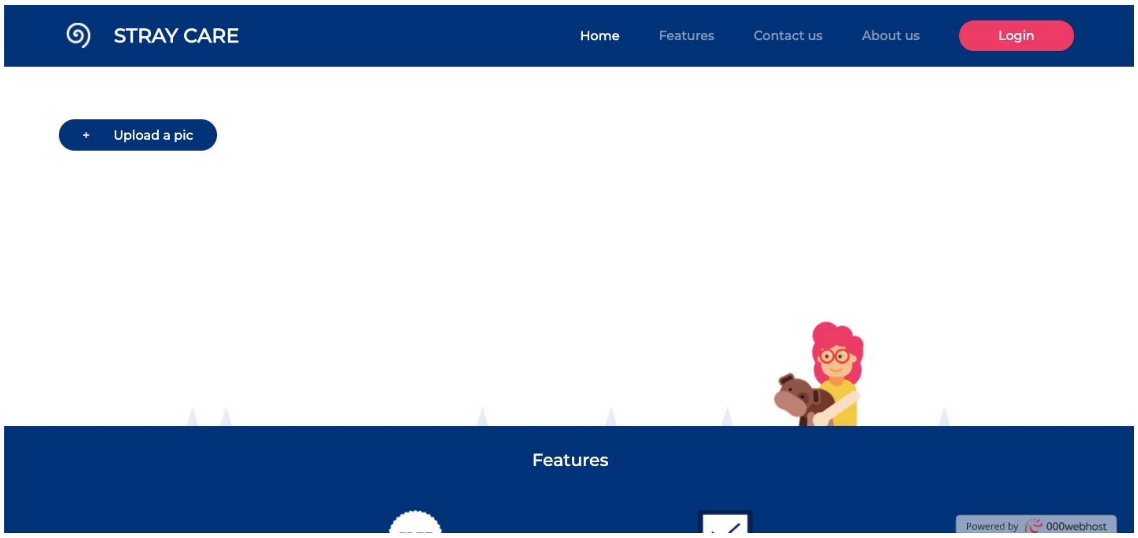
```
<style>
.parallax {
   /* The image used */
   background-image: url("img_parallax.jpg");
   /* Set a specific height */
   height: 500px;
   /* Create the parallax scrolling effect */
   background-attachment: fixed;
   background-position: center;
   background-repeat: no-repeat;
   background-size: cover;
}
</style>
<!-- Container element -->
<div class="parallax"></div>
```

The example above used pixels to set the height of the image. If you want to use percent, for example 100%, to make the image fit the whole screen, set the height of the parallax container to 100%.



```
/* LANDINGG */
.landing-main .landing-image {
  width: 100%;
  background-image: url("bg_image.png");
  min-height: 700px;
  background-attachment: fixed;
  background-position: center;
  background-repeat: no-repeat;
  background-size: cover;
}
.landing-main {
  width: 100%;
  position: relative;
  top: 80px;
}
```





Chapter 7

Embedding Maps and Video

Google Maps is a web mapping service developed by Google. It offers satellite imagery, street maps, 360° panoramic views of streets (Street View), real-time traffic conditions (Google Traffic), and route planning for traveling by foot, car, bicycle (in beta), or public transportation. OpenStreetMap is built by a community of mappers that contribute and maintain data about roads, trails, cafés, railway stations, and much more, all over the world. OpenStreetMap is open data: you are free to use it for any purpose as long as you credit OpenStreetMap and its contributors. Leaflet is the leading open-source JavaScript library for mobile-friendly interactive maps.

The <iframe> tag specifies an inline frame. An inline frame is used to embed another document within the current HTML document. The HTML <video> element is used to show a video on a web page.

Example

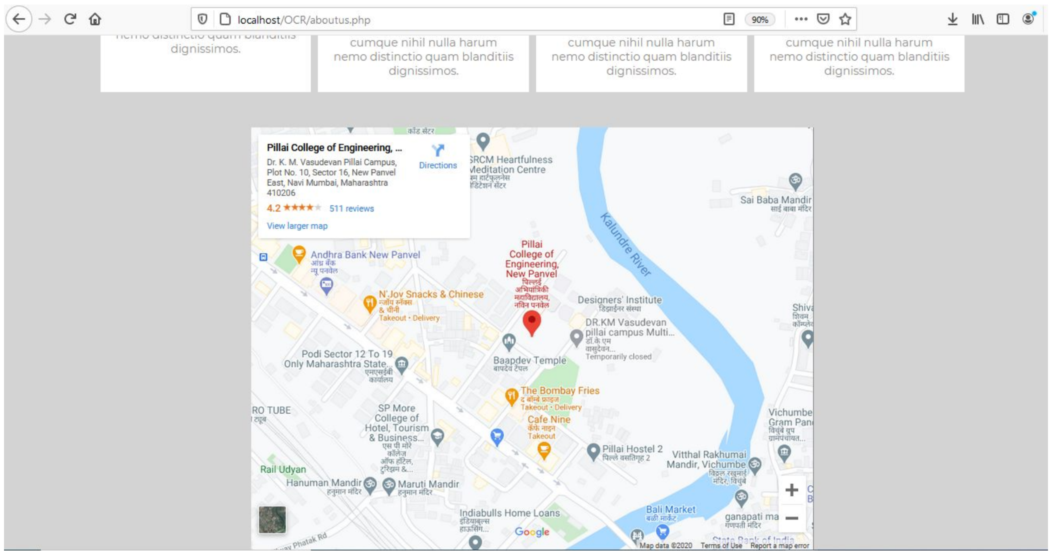
```
<video width="320" height="240" controls>

  <source src="movie.mp4" type="video/mp4">

  <source src="movie.ogg" type="video/ogg">

Your browser does not support the video tag.

</video>
```

The controls attribute adds video controls, like play, pause, and volume. It is a good idea to always include width and height attributes. If height and width are not set, the page might flicker while the video loads. The <source> element allows you to specify alternative video files which the browser may choose from. The browser will use the first recognized format. The text between the <video> and </video> tags will only be displayed in browsers that do not support the <video> element.

Code to Embed map using google:

```
<center>

  <iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3775.[illegible]!2d73.[illegible]!3d18.
  [illegible]!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.
  1!3m3!1m2!1s[illegible]%20College%20of%20Engineering%2C%20New%20Panvel!5e0!3m2!1sen!2sin!4v[illegible]
  !5m2!1sen!2sin" width="380" height="600" frameborder="0" style="border:0;" allowfullscreen="" aria-hidden="false" tabindex="0"></iframe>

</center>
```

Output :


localhost/OCR/aboutus.php

nemo distinctio quam blanditiis dignissimos.

cumque nihil nulla harum nemo distinctio quam blanditiis dignissimos.

cumque nihil nulla harum nemo distinctio quam blanditiis dignissimos.

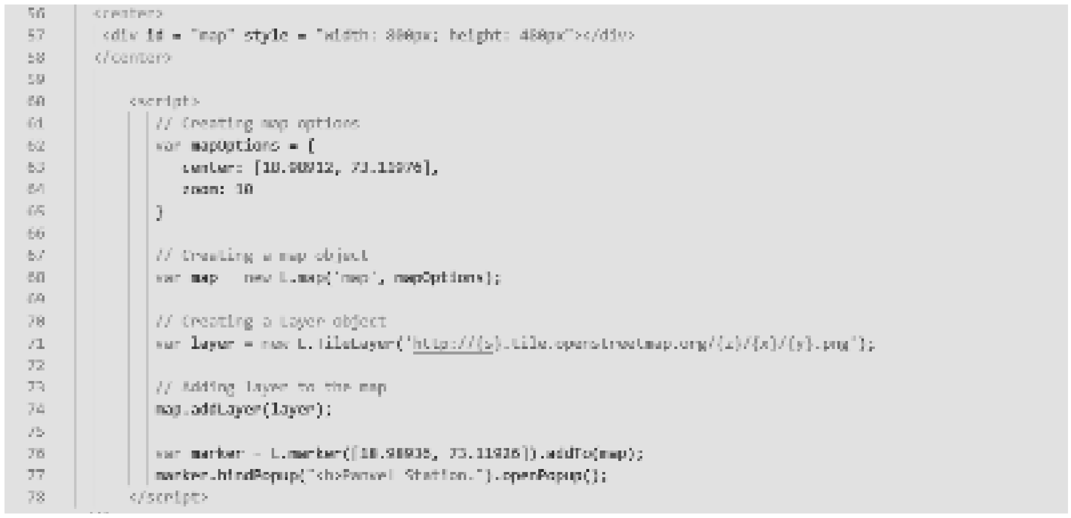
cumque nihil nulla harum nemo distinctio quam blanditiis dignissimos.

Pillai College of Engineering, ...
Dr. K. M. Vasudevan Pillai Campus, Plot No. 10, Sector 16, New Panvel East, Navi Mumbai, Maharashtra 410206
4.2 ★★★★ 511 reviews
View larger map

Directions
SRCM Heartfulness Meditation Centre
Sai Baba Mandir
Kalundre River
Andhra Bank New Panvel
Pillai College of Engineering, New Panvel
N'Joy Snacks & Chinese
Designers' Institute
DR.KM Vasudevan pillai campus Multi...
Temporarily closed
Podi Sector 12 To 19 Only Maharashtra State...
Baapdev Temple
The Bombay Fries
SP More College of Hotel, Tourism & Business...
Cafe Nine
Pillai Hostel 2
Vitthal Rakhumai Mandir, Vichumbe
Vichumbe Gram Pan
Rail Udyan
Hanuman Mandir
Maruti Mandir
Indiabulls Home Loans
Bali Market
ganapati ma
Google
Map data ©2020 Terms of Use Report a map error

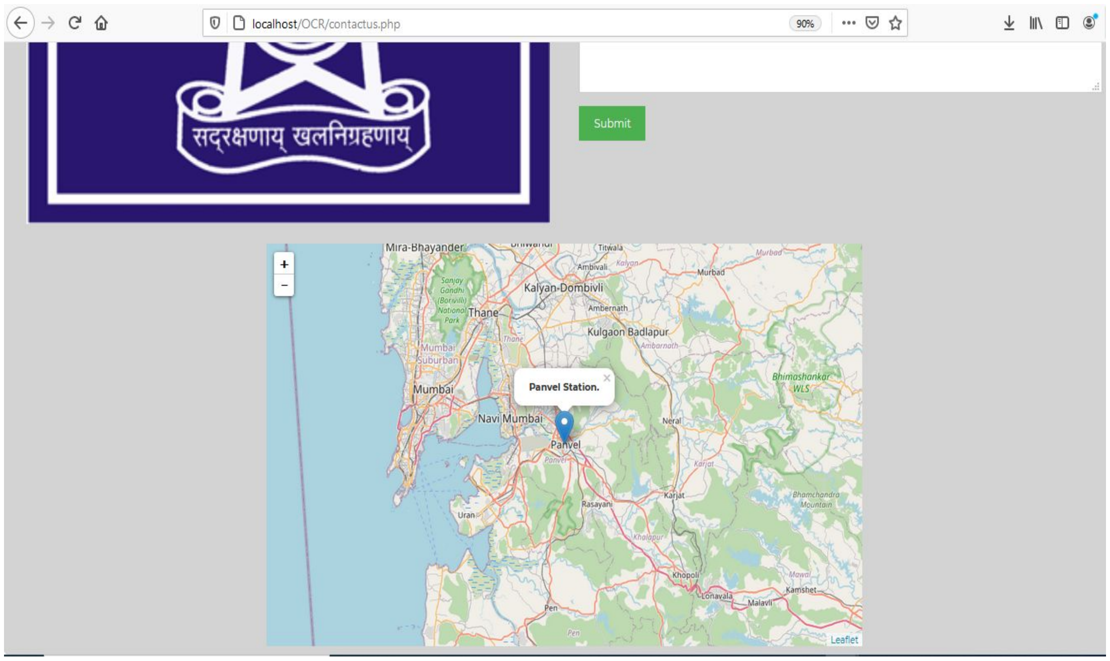
Code to Embed map using leaflet:

```
<head>
  <link rel="stylesheet" href="media/css/main/contactus.css" />
  <link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css"
  integrity="sha512-xodZBNTC5n17Xt2atTPuE1HxjVMSvLVW9ocqUKLsCC5CXdbqCmblAshOMAS6/keqq/sMZMZ19scR4PsZChSR7A=="
  crossorigin=""/>
  <script src = "http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.js"></script>
</head>
```



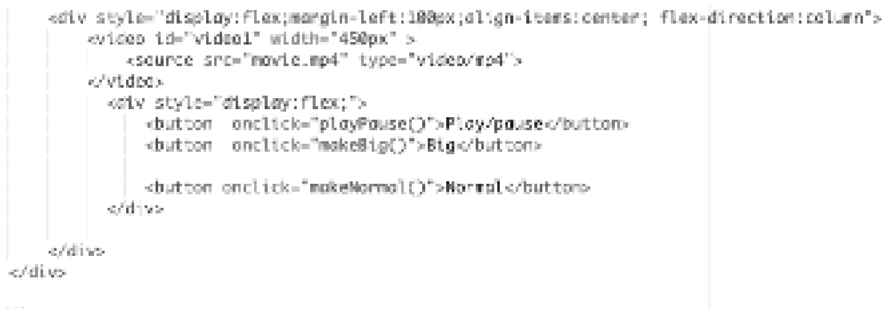
```
<center>
 <div id = "map" style = "width: 800px; height: 400px"></div>
</center>

    <script>
       // Creating map options
       var mapOptions = {
          center: [18.98912, 73.11976],
          zoom: 10
       }

       // Creating a map object
       var map = new L.map('map', mapOptions);

       // Creating a Layer object
       var layer = new L.TileLayer('http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png');

       // Adding layer to the map
       map.addLayer(layer);

       var marker = L.marker([18.98935, 73.11926]).addTo(map);
       marker.bindPopup("<b>Panvel Station.").openPopup();
    </script>
```

Output :

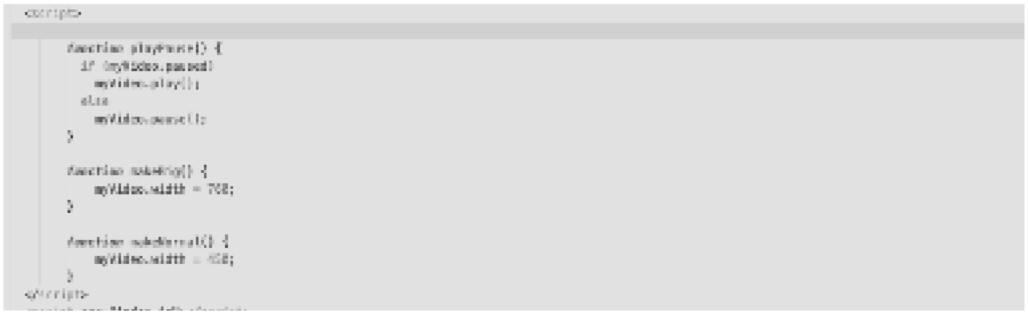


localhost/OCR/contactus.php

Code to embed Video:

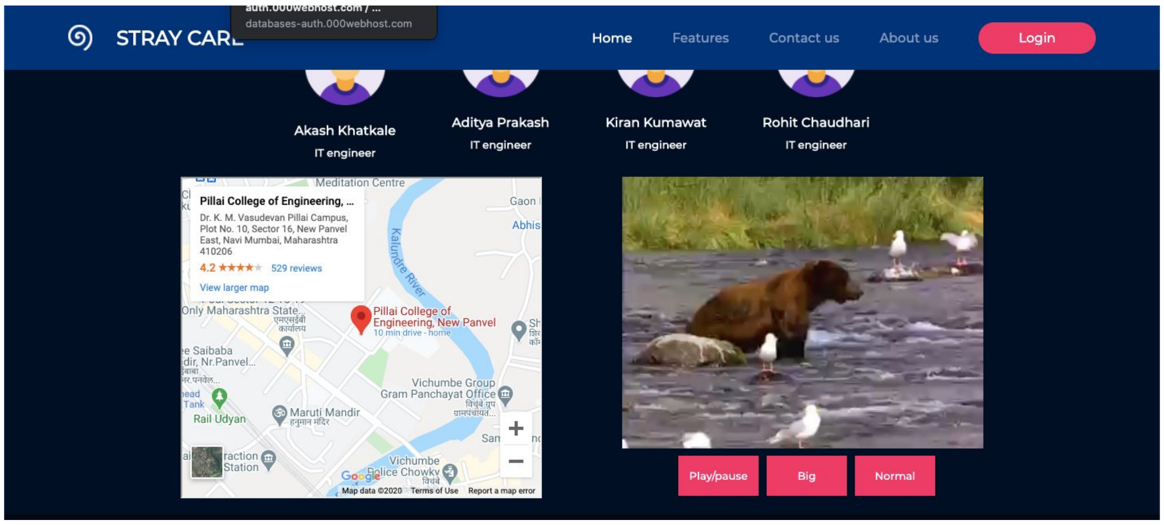


```
    <div style="display:flex;margin-left:100px;align-items:center; flex-direction:column">
        <video id="video1" width="450px" >
            <source src="movie.mp4" type="video/mp4">
        </video>
          <div style="display:flex;">
              <button  onclick="playPause()">Play/pause</button>
              <button  onclick="makeBig()">Big</button>

              <button onclick="makeNormal()">Normal</button>
          </div>

    </div>
</div>
```



```
<script>

    function playPause() {
      if (myVideo.paused)
        myVideo.play();
      else
        myVideo.pause();
    }

    function makeBig() {
        myVideo.width = 700;
    }

    function makeNormal() {
        myVideo.width = 450;
    }
</script>
[illegible]
```

Output


auth.000webhost.com / ...
databases-auth.000webhost.com
STRAY CARE
Home
Features
Contact us
About us
Login
Akash Khatkale
IT engineer
Aditya Prakash
IT engineer
Kiran Kumawat
IT engineer
Rohit Chaudhari
IT engineer
Pillai College of Engineering, ...
Dr. K. M. Vasudevan Pillai Campus, Plot No. 10, Sector 16, New Panvel East, Navi Mumbai, Maharashtra 410206
4.2 ★★★★★ 529 reviews
View larger map
Pillai College of Engineering, New Panvel
Kalundre River
Vichumbe Group Gram Panchayat Office
Maruti Mandir
Rail Udyan
Map data ©2020 Terms of Use Report a map error
Play/pause
Big
Normal

Chapter 8

HTML5 based form validation

Forms are used in webpages for the user to enter their required details that are further send it to the server for processing. A form is also known as web form or HTML form. Form validation helps us to ensure that users fill out forms in the correct format, making sure that submitted data will work successfully with our applications.

Go to any popular site with a registration form, and you will notice that they give you feedback when you don't enter your data in the format they are expecting. You'll get messages such as:

"This field is required" (you can't leave this field blank)

"Please enter your phone number in the format xxx-xxxx" (it enforces three numbers followed by a dash, followed by four numbers)

"Please enter a valid e-mail address" (if your entry is not in the format of "somebody@example.com")

"Your password needs to be between 8 and 30 characters long, and contain one uppercase letter, one symbol, and a number"

This is called form validation — when you enter data, the web application checks it to see that the data is correct. If correct, the application allows the data to be submitted to the server and (usually) saved in a database; if not, it gives you an error message explaining what corrections need to be made. Form validation can be implemented in a number of different ways.

We want to make filling out web forms as easy as possible. So why do we insist on validating our forms? There are three main reasons:

We want to get the right data, in the right format — our applications won't work properly if our user's data is stored in the incorrect format, or if they don't enter the correct information, or omit information altogether.

We want to protect our users' accounts — by forcing our users to enter secure passwords, it makes it easier to protect their account information.

We want to protect ourselves — there are many ways that malicious users can misuse unprotected forms to damage the application they are part of (see Website security).

Different types of form validation

There are two different types of form validation which you'll encounter on the web:

Client-side validation is validation that occurs in the browser before the data has been submitted to the server. This is more user-friendly than server-side validation as it gives an instant response. This can be further subdivided:

JavaScript validation is coded using JavaScript. It is completely customizable.

Built-in form validation using HTML5 form validation features. This generally does not require JavaScript. Built-in form validation has better performance, but it is not as customizable as JavaScript.

Server-side validation is validation which occurs on the server after the data has been submitted. Server-side code is used to validate the data before it is saved into the database. If the data fails authentication, a response is sent back to the client to tell the user what corrections to make. Server-side validation is not as user-friendly as client-side validation, as it does not provide errors until the entire form has been submitted. However, server-side validation is your application's last line of defence against incorrect or even malicious data. All popular server-side frameworks have features for validating and sanitizing data (making it safe).

1. Specialized Input Types

HTML5 introduced several new input types. They can be used to create input boxes, which will accept only a specified kind of data.

The new input types are as follows:

Color, date, datetime, email , month .number, range . search, tel , time , url , week

To use one of the new types, include them as the value of the type attribute:

```
<input type="email"/>
```

2. Required Fields

By simply adding the "required" attribute to a <input>, <select> or <textarea>, you tell the browser that a value must be provided in this field. Think of this as the red asterisk* we see in most registration forms.

```
<input type="checkbox" name="terms" required >
```

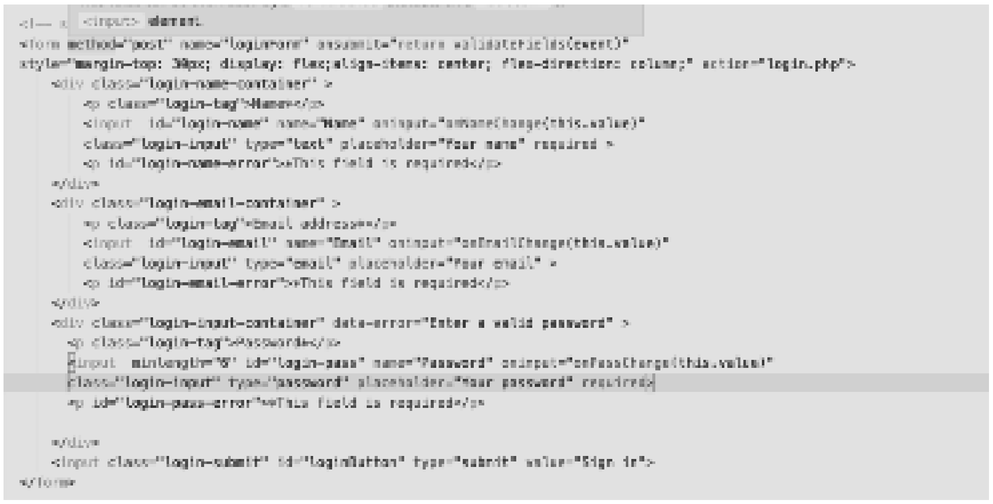
3. Limits

We can set some basic limitations like max length and minimum and maximum values for number fields. To limit the length of input fields and textareas, use the "maxlength" attribute. What this does is to forbid any string longer than the field's "maxlength" value to be entered at all. If you try and paste a string witch exceeds this limit, the form will simply clip it.

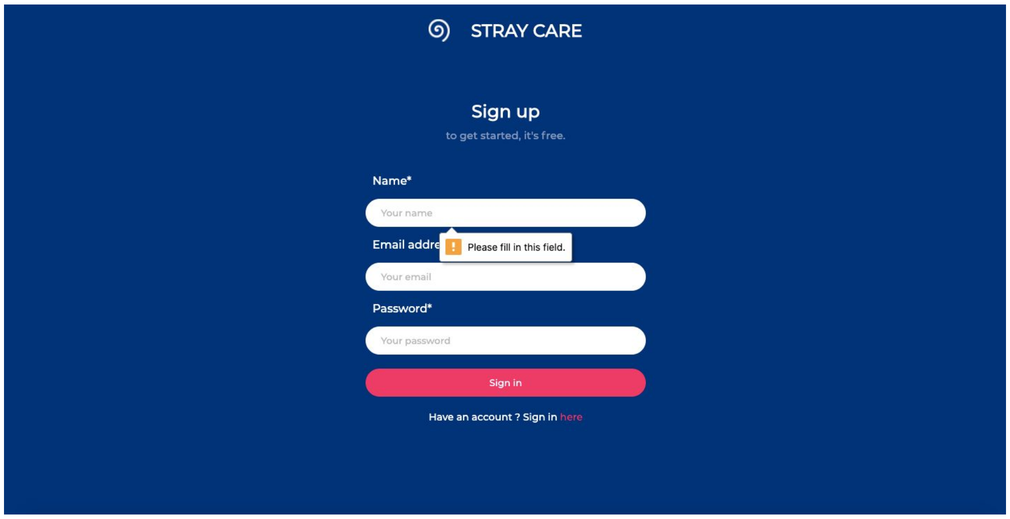
```
<input type="text" name="name" required  maxlength="15">
```

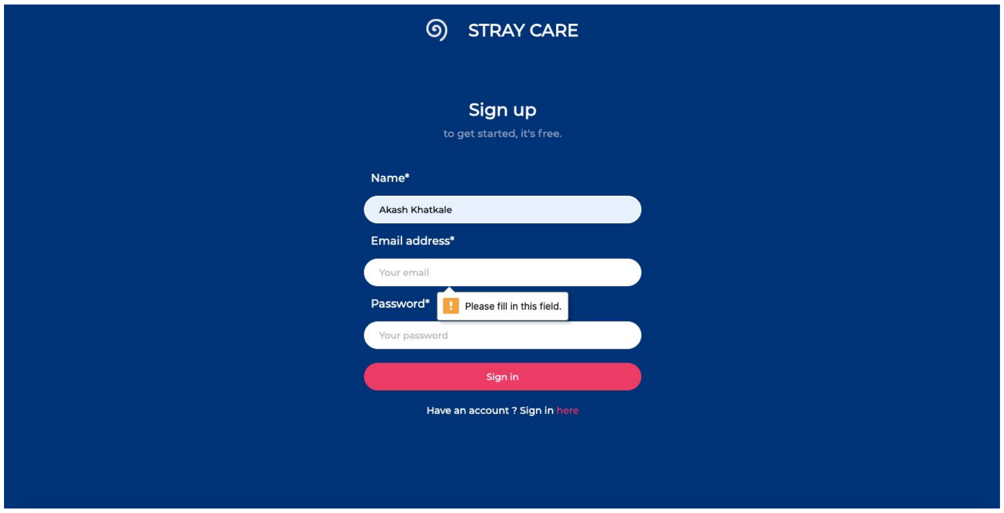
The <input type="number"> fields use "max" and "min" attributes to create a range of possible values - in our example we've made the minimum allowed age to be 18 (too bad you can be whatever age you want on the internet).

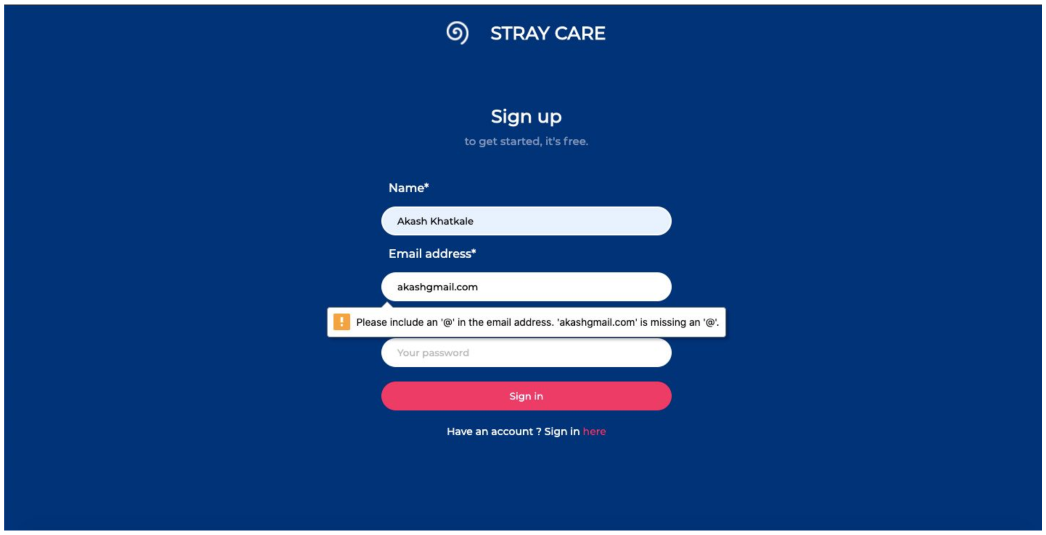
```
<input type="number" name="age" min="18" required>
```



```
<!-- [illegible] <input> element.
<form method="post" name="loginForm" onsubmit="return validateFields(event)"
style="margin-top: 30px; display: flex;align-items: center; flex-direction: column;" action="login.php">
    <div class="login-name-container" >
        <p class="login-tag">Name*</p>
        <input  id="login-name" name="Name" oninput="onNameChange(this.value)"
        class="login-input" type="text" placeholder="Your name" required >
        <p id="login-name-error">*This field is required</p>
    </div>
    <div class="login-email-container" >
        <p class="login-tag">Email address*</p>
        <input  id="login-email" name="Email" oninput="onEmailChange(this.value)"
        class="login-input" type="email" placeholder="Your email" >
        <p id="login-email-error">*This field is required</p>
    </div>
    <div class="login-input-container" data-error="Enter a valid password" >
      <p class="login-tag">Password*</p>
      <input  minlength="8" id="login-pass" name="Password" oninput="onPassChange(this.value)"
      class="login-input" type="password" placeholder="Your password" required>
      <p id="login-pass-error">*This field is required</p>

    </div>
    <input class="login-submit" id="loginButton" type="submit" value="Sign in">
</form>
```





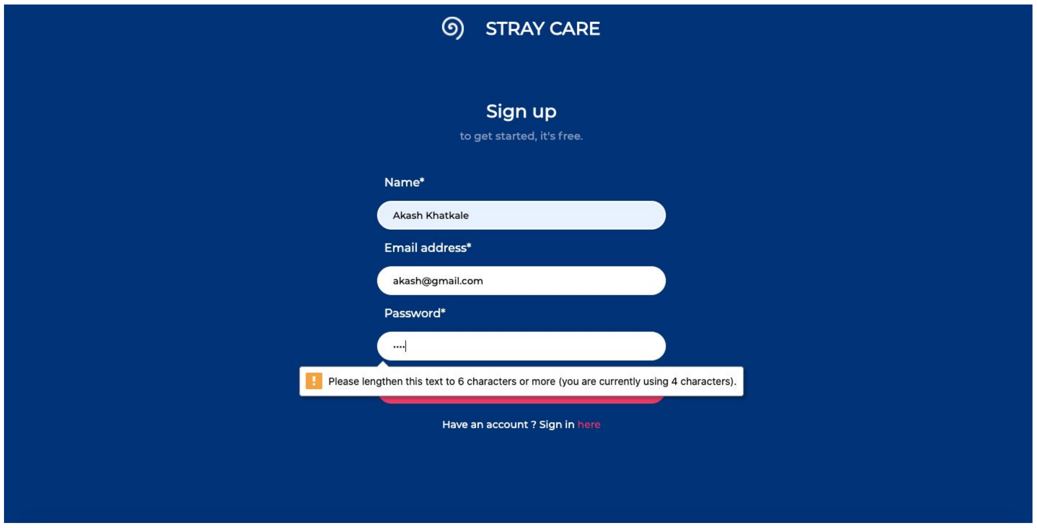


Akash Khatkale

akashgmail.com

Please include an '@' in the email address. 'akashgmail.com' is missing an '@'.

Your password



Akash Khatkale

akash@gmail.com

Please lengthen this text to 6 characters or more (you are currently using 4 characters).

Chapter 9

Javascript based form validation

Validating form input with JavaScript is easy to do and can save a lot of unnecessary calls to the server as all processing is handled by the web browser. It can prevent people from leaving fields blank, from entering too little or too much or from using invalid characters.

Forms validation on the client-side is essential — it saves time and bandwidth, and gives you more options to point out to the user where they've gone wrong in filling out the form. Having said that, I don't mean that you don't need server-side validation. People who visit your site may use an old browser or have JavaScript disabled, which will break client-only validation. Client and server-side validation complement each other, and as such, they really shouldn't be used independently.

Why is Client Side Validation Good?

There are two good reasons to use client-side validation:

1. It's a fast form of validation: if something's wrong, the alarm is triggered upon submission of the form.
2. You can safely display only one error at a time and focus on the wrong field, to help ensure that the user correctly fills in all the details you need.

Two Major Validation Approaches

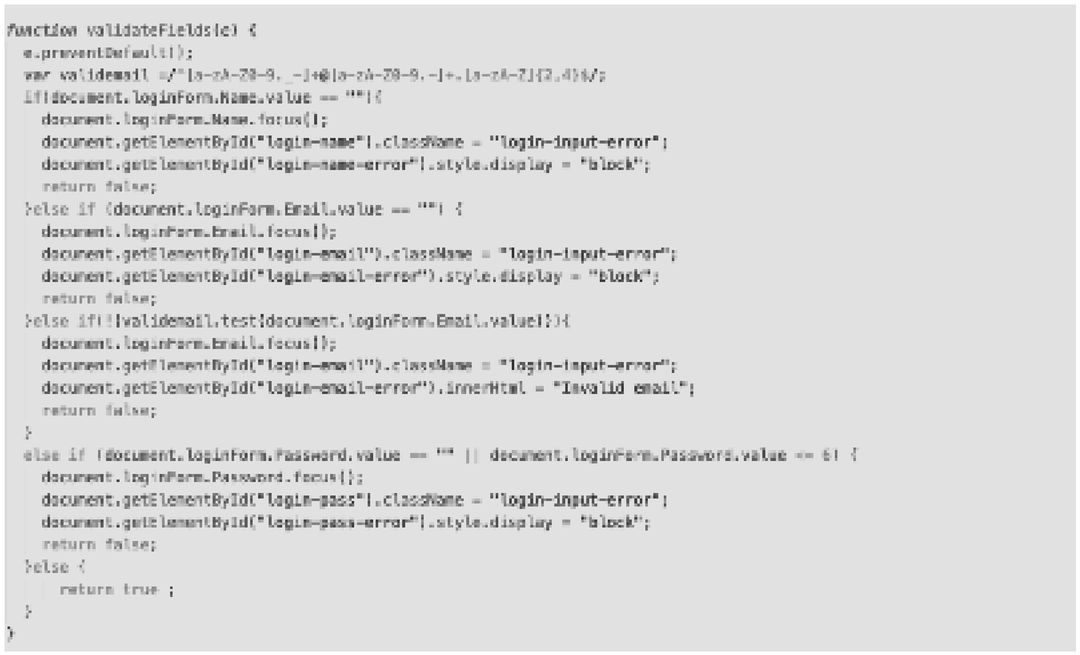
1. Display the errors one by one, focusing on the offending field
2. Display all errors simultaneously, server-side validation style

While displaying all errors simultaneously is required for server-side validation, the better method for validation on the client-side is to show one error at a time. This makes it possible to highlight only the field that has been incorrectly completed, which in turn makes revising and successfully submitting the form much easier for the visitor. If you present users with all errors at the same time, most people will try to remember and correct them at once, instead of attempting to re-submit after each correction.

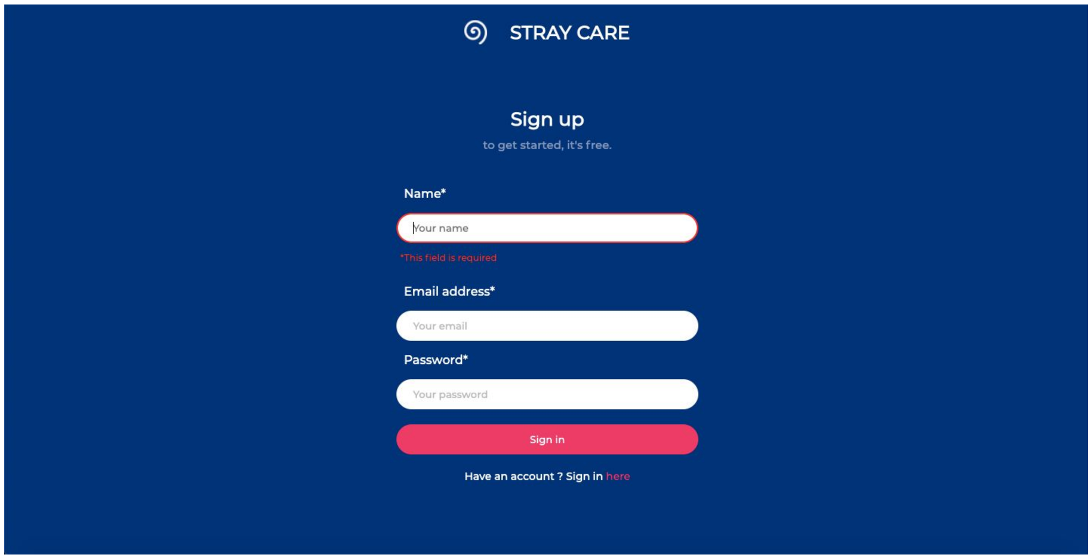
```
 function validateForm() {

   var x = document.forms["myForm"]["fname"].value;

   if (x == "") {

       alert("Name must be filled out");

       return false;

   }

}

<form    name="myForm"    action="/action_page.php"    onsubmit="return    validateForm()"
method="post">

Name: <input type="text" name="fname">

<input type="submit" value="Submit">

</form>

function checkpassword(pform1){

var str=pform1.password.value;
//check required fields
//password should be minimum 4 chars but not greater than 8
if ((str.length < 4) || (str.length > 8)) {
```

```
function checkpassword(pform1){
var str=pform1.password.value;

//check required fields

//password should be minimum 4 chars but not greater than 8

if ((str.length < 4) || (str.length > 8)) {
```

```
alert("Invalid password length.")
pform1.password.focus()
return false
}

}
function checkemailphone(pform1){
var email = pform1.email.value;
var phone = pform1.phone.value;
var cleanstr = phone.replace(/[().- ]/g, '');
var validemail =/^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+.[a-zA-Z]{2,4}$/;
if(!(validemail.test(email))){
alert("Invalid email address")
pform1.email.focus()
return false
}
//check phone number
if (isNaN(parseInt(cleanstr))) {
alert("The phone number contains unwanted characters.")
}
}
```



```
function validateFields(e) {
  e.preventDefault();
  var validemail =/^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+.[a-zA-Z]{2,4}$/;
  if(document.loginForm.Name.value == ""){
    document.loginForm.Name.focus();
    document.getElementById("login-name").className = "login-input-error";
    document.getElementById("login-name-error").style.display = "block";
    return false;
  }else if (document.loginForm.Email.value == "") {
    document.loginForm.Email.focus();
    document.getElementById("login-email").className = "login-input-error";
    document.getElementById("login-email-error").style.display = "block";
    return false;
  }else if(!(validemail.test(document.loginForm.Email.value))){
    document.loginForm.Email.focus();
    document.getElementById("login-email").className = "login-input-error";
    document.getElementById("login-email-error").innerHtml = "Invalid email";
    return false;
  }
  else if (document.loginForm.Password.value == "" || document.loginForm.Password.value <= 6) {
    document.loginForm.Password.focus();
    document.getElementById("login-pass").className = "login-input-error";
    document.getElementById("login-pass-error").style.display = "block";
    return false;
  }else {
      return true ;
  }
}
```



STRAY CARE

Sign up

to get started, it's free.

Name*

Your name

*This field is required

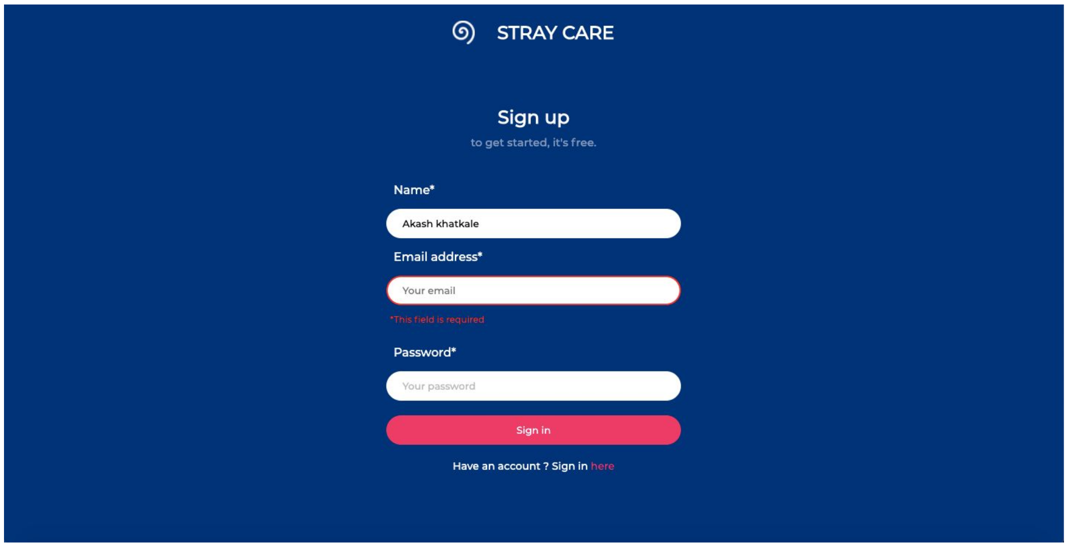
Email address*

Your email

Password*

Your password

Have an account ? Sign in here


STRAY CARE

Sign up

to get started, it's free.

Name*

Akash khatkale

Email address*

Your email

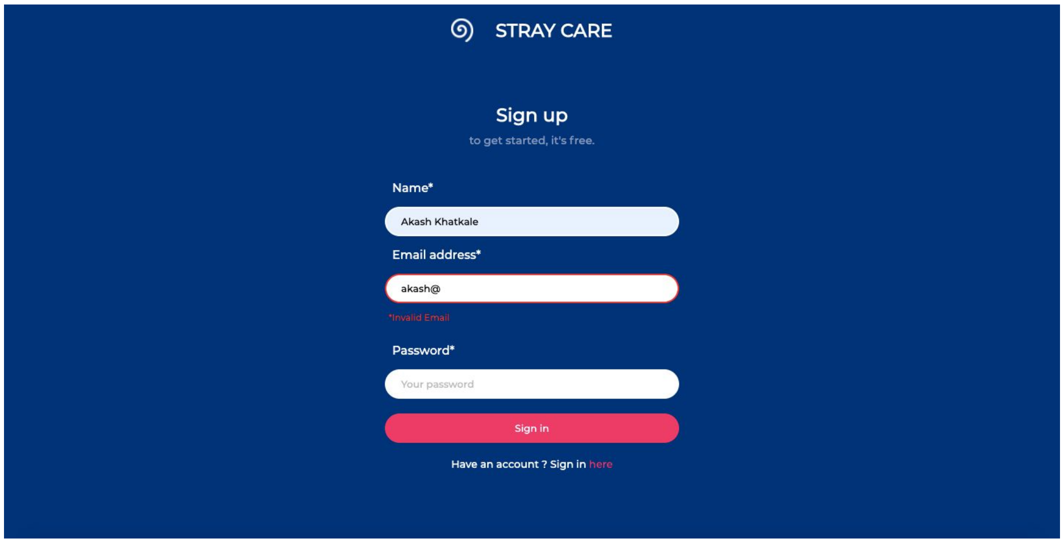
*This field is required

Password*

Your password

Sign in

Have an account ? Sign in here


STRAY CARE

Sign up

to get started, it's free.

Name*

Akash Khatkale

Email address*

akash@

*Invalid Email

Password*

Your password

Sign in

Have an account ? Sign in here

Chapter 10

Server side programming using PHP

PHP stands for Hypertext Preprocessor. PHP is a very popular and widely-used open source server-side scripting language to write dynamically generated web pages. PHP was originally created by Rasmus Lerdorf in 1994. It was initially known as Personal Home Page.

PHP scripts are executed on the server and the result is sent to the web browser as plain HTML. PHP can be integrated with the number of popular databases, including MySQL, PostgreSQL, Oracle, Microsoft SQL Server, Sybase, and so on.

What You Can Do with PHP

- There are lot more things you can do with PHP.
- You can generate pages and files dynamically.
- You can create, open, read, write and close files on the server.
- You can collect data from a web form such as user information, email, phone no, etc.
- You can send emails to the users of your website.
- You can send and receive cookies to track the visitor of your website.
- You can store, delete, and modify information in your database.
- You can restrict unauthorized access to your website.
- You can encrypt data for safe transmission over internet.
- The list does not end here, there are many other interesting things that you can do with PHP. You will learn about all of them in detail in upcoming chapters.

Advantages of PHP over Other Languages

- If you're familiar with other server-side languages like ASP.NET or Java, you might be wondering what makes PHP so special. There are several advantages why one should choose PHP.
- Easy to learn: PHP is easy to learn and use. For beginner programmers who just started out in web development, PHP is often considered as the preferable choice of language to learn.
- Open source: PHP is an open-source project. It is developed and maintained by a worldwide community of developers who make its source code freely available to

download and use.

- Portability: PHP runs on various platforms such as Microsoft Windows, Linux, Mac OS, etc. and it is compatible with almost all servers used today such Apache, IIS, etc.
- Fast Performance: Scripts written in PHP usually execute or runs faster than those written in other scripting languages like ASP, Ruby, Python, Java, etc.
- Vast Community: Since PHP is supported by the worldwide community, finding help or documentation related to PHP online is extremely easy.

Setting Up a Local Web Server

PHP script execute on a web server running PHP. So before you start writing any PHP program you need the following program installed on your computer.

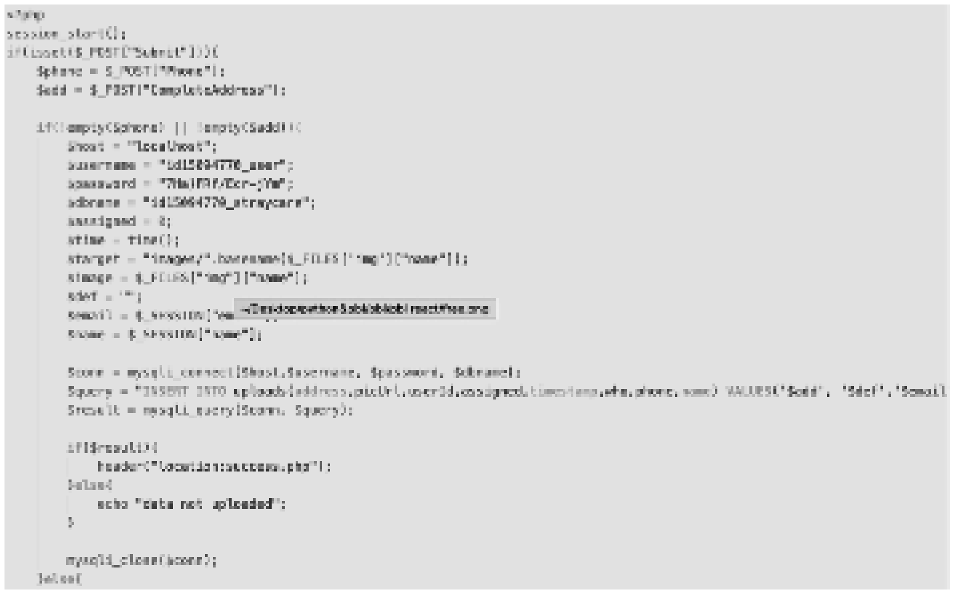
1. The Apache Web server
2. The PHP engine
3. The MySQL database server

You can either install them individually or choose a pre-configured package for your operating system like Linux and Windows. Popular pre-configured package are XAMPP and WampServer.

Creating Your First PHP Script

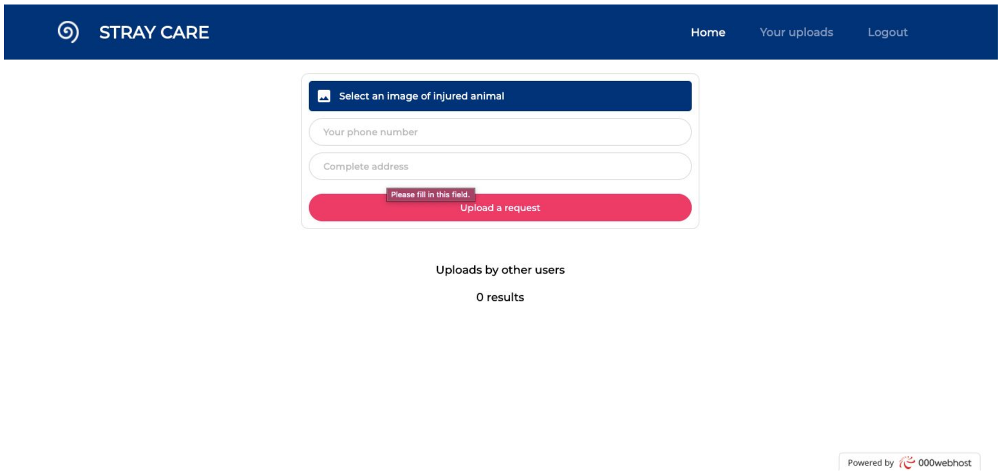
```
<!DOCTYPE HTML>
<html>
<head>
  <title>PHP Application</title>
</head>
<body>
<?php
// Display greeting message
echo 'Hello World!';
?>
</body>
</html>
```

Code:

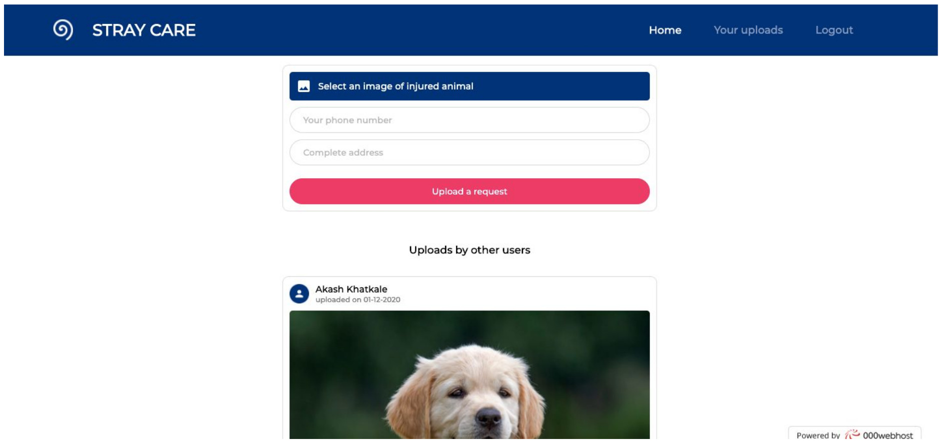


```
<?php
session_start();
if(isset($_POST["Submit"])){
    $phone = $_POST["Phone"];
    $add = $_POST["CompleteAddress"];

    if(!empty($phone) || !empty($add)){
        $host = "localhost";
        $username = "id15064770_user";
        $password = "7Ma!F9f/Ecr-jYm";
        $dbname = "id15064770_straycare";
        $assigned = 0;
        $time = time();
        $target = "images/".basename($_FILES['img']["name"]);
        $image = $_FILES["img"]["name"];
        $def = '';
        $email = $_SESSION["em[illegible]
        $name = $_SESSION["name"];

        $conn = mysqli_connect($host,$username, $password, $dbname);
        $query = "INSERT INTO uploads(address,picUrl,userId,assigned,timestamp,who,phone,name) VALUES('$add', '$def','$email
        $result = mysqli_query($conn, $query);

        if($result){
            header("location:success.php");
        }else{
            echo "data not uploaded";
        }

        mysqli_close($conn);
    }else{
```

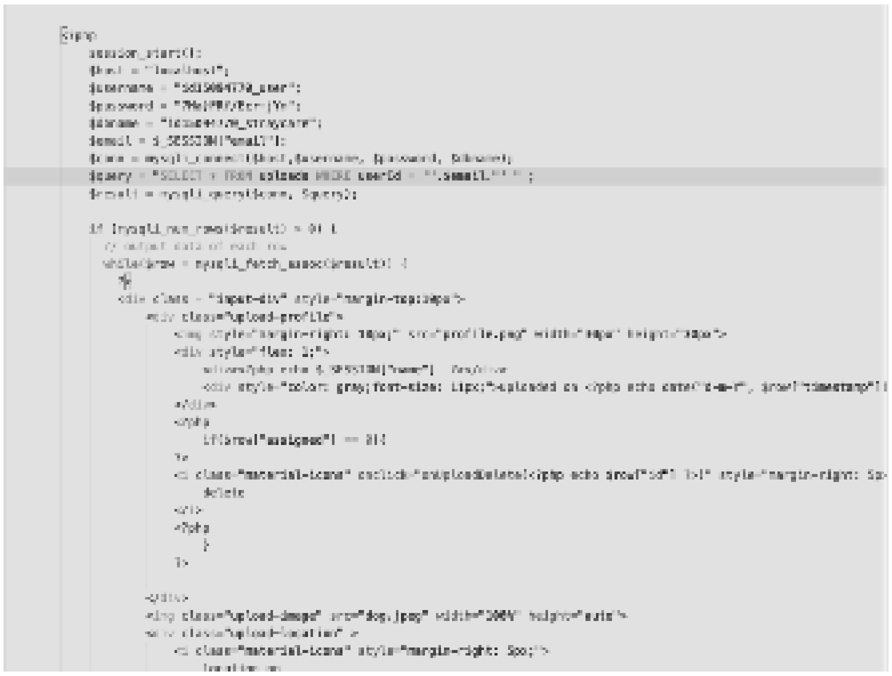
OUTPUT Before:



OUTPUT after:

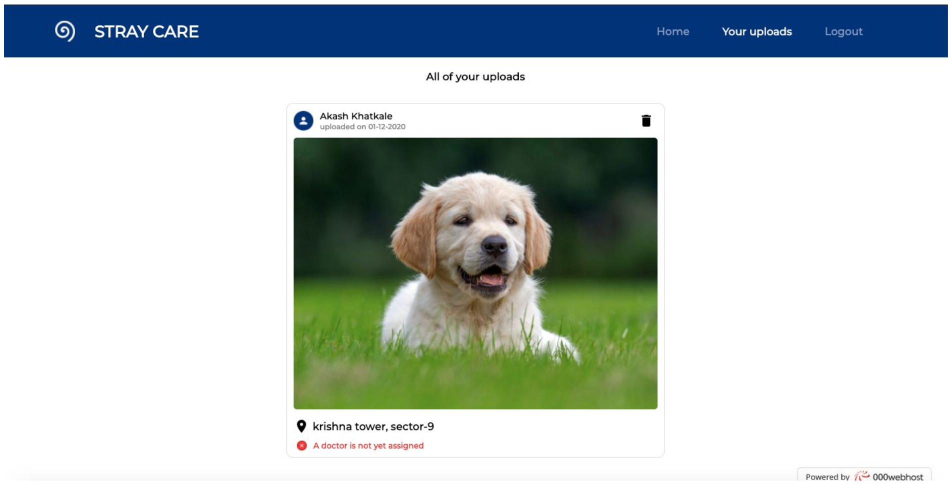


CODE:



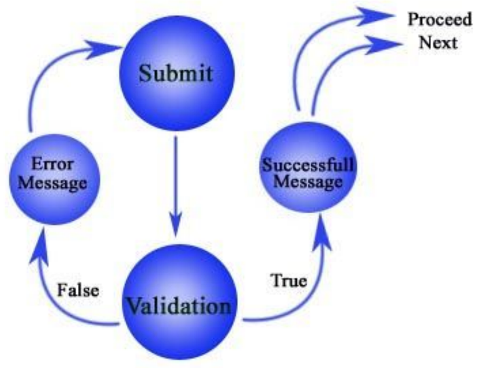
```
<?php
  session_start();
  $host = "localhost";
  $username = "id15064739_user";
  $password = "[illegible]";
  $dbname = "id15064739_straycare";
  $email = $_SESSION["email"];
  $conn = mysqli_connect($host,$username, $password, $dbname);
  $query = "SELECT * FROM uploads WHERE userId = '".$email."' " ;
  $result = mysqli_query($conn, $query);

  if (mysqli_num_rows($result) > 0) {
    // output data of each row
    while($row = mysqli_fetch_assoc($result)) {
      ?>
      <div class = "input-div" style="margin-top:30px">
          <div class="upload-profile">
              <img style="margin-right: 10px;" src="profile.png" width="40px" height="40px">
              <div style="flex: 1;">
                  <b><?php echo $_SESSION["name"]; ?></b>
                  <div style="color: gray;font-size: 11px;">uploaded on <?php echo date("d-m-Y", $row["timestamp"]
              </div>
              <?php
                  if($row["assigned"] == 0){
              ?>
              <i class="material-icons" onclick="onUploadDelete(<?php echo $row["id"] ?>)" style="margin-right: 5p
                  delete
              </i>
              <?php
                  }
              ?>

          </div>
          <img class="upload-image" src="dog.jpeg" width="100%" height="auto">
          <div class="upload-location" >
              <i class="material-icons" style="margin-right: 5px;">
                  location_on
```

OUTPUT:



Chapter 11
PHP form validation

It is very essential to have the input to your form validated before taking the form submission data for further processing. When there are many fields in the form, the PHP validation script becomes too complex.



Validation rules:

- Name : Should required letters and white-spaces
- Email : Should required @ and .
- Website : Should required a valid URL
- Radio : Must be selectable at least once
- Check Box : Must be checkable at least once
- Drop Down menu : Must be selectable at least once

<u>Validation for non-empty, alphabets and whitespace only</u>

The following code is added within the form

```
<label>Full name<span class="note">*</span>:</label>
  <input type="text" name="full_name" placeholder="FirstName LastName"
autofocus="autofocus" value="<?php echo $_POST['full_name']; ?>">
  <?php echo "<p class='note'>".$msg_name."</p>";?>
  <?php echo "<p class='note'>".$msg2_name."</p>";?>
```

Code for validation

```
if (isset($_POST['submit'])) {
//checking name
if(empty($_POST['full_name']))
$msg_name = "You must supply your name";
$name_subject = $_POST['full_name'];
$name_pattern = '/^[a-zA-Z ]*$/';
preg_match($name_pattern, $name_subject, $name_matches);
if(!$name_matches[0])
$msg2_name = "Only alphabets and white space allowed";
}
```

<u>email Validation</u>

Code added within the form

```
<label>Email address<span class="note">*</span>:</label>
  <input type="text" name="email_addr" value="<?php echo $_POST['email_addr']; ?>">
   <?php echo "<p class='note'>".$msg_email."</p>";?>
  <?php echo "<p class='note'>".$msg2_email."</p>";?>
```

Code for validation

```
if (isset($_POST['submit'])) {
//check email
if(empty($_POST['email_addr']))
$msg_email = "You must supply your email";
$email_subject = $_POST['email_addr'];
$email_pattern = '/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/';
preg_match($email_pattern, $email_subject, $email_matches);
if(!$email_matches[0])
$msg2_email = "Must be of valid email format";
}
```

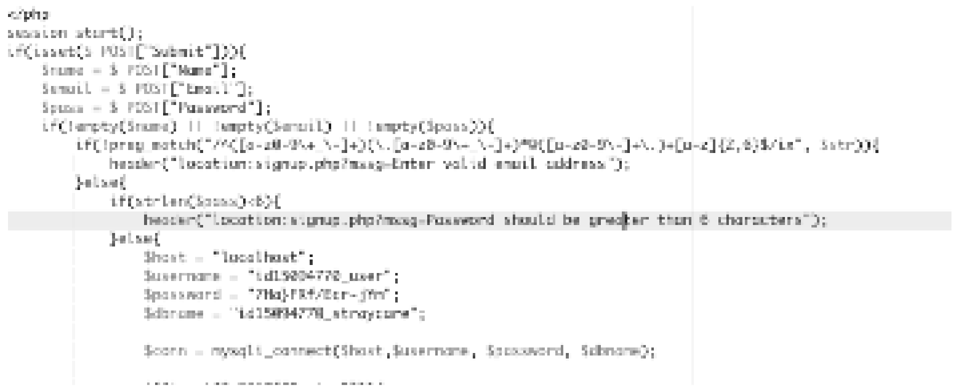
Selection list Validation

Code added within the form

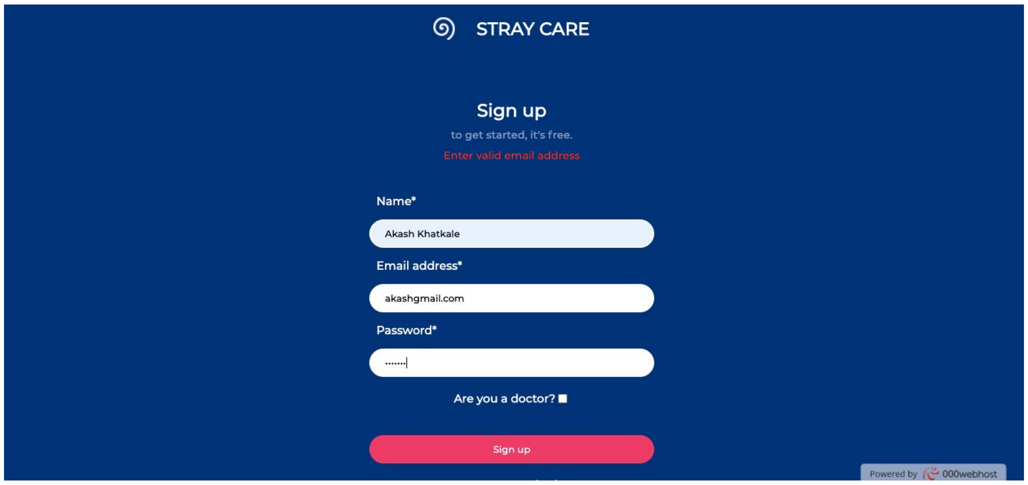
```
<label>Select Tour Package<span class="note">*</span>:</label>
  <select name="package">
        <option value="Goa" <?= ($_POST['package'] == "1")? "selected":"";?>>Goa</options>
        <option value="Kashmir" <?= ($_POST['package'] == "2")?
"selected":"";?>>Kashmir</options>
        <option value="Rajasthan" <?= ($_POST['package'] == "3")?
"selected":"";?>>Rajasthan</options>
  </select>
```

Code for validation

```
if (isset($_POST['submit'])) {
if(empty($_POST['package']))
$msg_package = "You must select a package";
}
```



```
<?php
session_start();
if(isset($_POST["Submit"])){
    $name = $_POST["Name"];
    $email = $_POST["Email"];
    $pass = $_POST["Password"];
    if(!empty($name) || !empty($email) || !empty($pass)){
        if(!preg_match("/^([a-z0-9\+_\-]+)(\.[a-z0-9\+_\-]+)*@([a-z0-9\-]+\.)+[a-z]{2,6}$/ix", $str)){
            header("location:signup.php?msg=Enter valid email address");
        }else{
            if(strlen($pass)<6){
                header("location:signup.php?msg=Password should be greater than 6 characters");
            }else{
                $host = "localhost";
                $username = "id15094770_user";
                $password = "7Hq}FR4/Ecr-jfm";
                $dbname = "id15094770_straycare";

                $conn = mysqli_connect($host,$username, $password, $dbname);
```


STRAY CARE

Sign up

to get started, it's free.

Enter valid email address

Name*

Akash Khatkale

Email address*

akashgmail.com

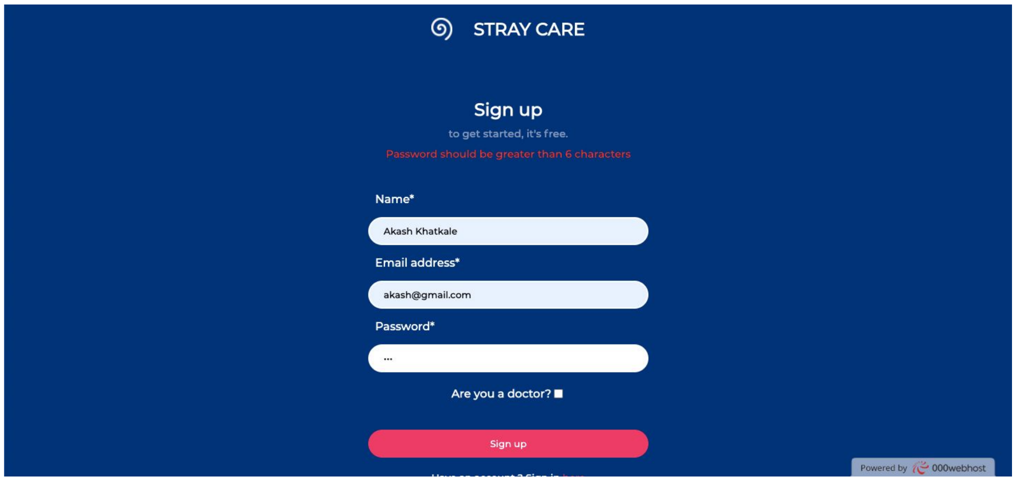
Password*

•••••••

Are you a doctor?

Sign up

Powered by 000webhost


STRAY CARE

Sign up

to get started, it's free.

Password should be greater than 6 characters

Name*

Akash Khatkale

Email address*

akash@gmail.com

Password*

•••

Are you a doctor?

Sign up

Powered by 000webhost

Chapter 12

PHP MySQL database operations

With PHP, you can connect to and manipulate databases MySQL is the most popular database system used with PHP.

What is MySQL?

- MySQL is a database system used on the web
- MySQL is a database system that runs on a server
- MySQL is ideal for both small and large applications
- MySQL is very fast, reliable, and easy to use
- MySQL uses standard SQL
- MySQL compiles on a number of platforms
- MySQL is free to download and use

The data in a MySQL database are stored in tables. A table is a collection of related data, and it consists of columns and rows. Databases are useful for storing information categorically. A company may have a database with the following tables:

- Employees
- Products
- Customers
- Orders

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
// Create connection
$conn = mysqli_connect($servername, $username, $password);
// Check connection
if (!$conn) {
   die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully";
?>
```

If you want to use PHP to query your MySQL database you can do that by either entering the MySQL query command in the PHP script or define the command as a variable and use the variable when needed

```
mysqli_query($query);
```

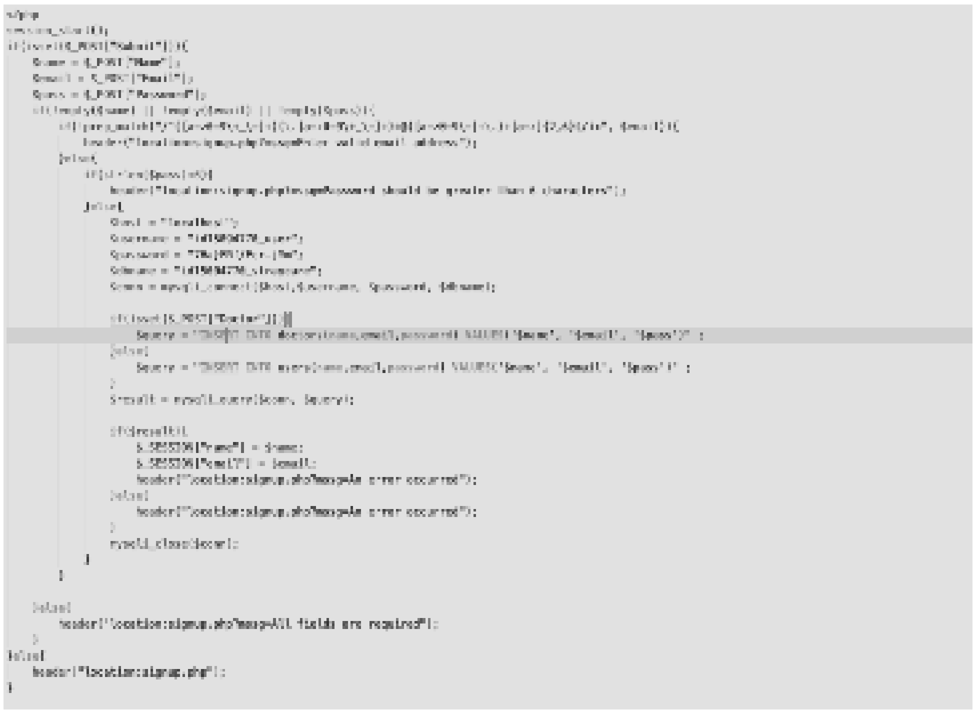
The command can be repeated again in the source code. All you need to do is to change the $query variable.

For example, here is the complete code that could be used to create a MySQL table in PHP:

```
<?php
$username = "your_username";
$password = "your_password";
$database = "your_database";
$mysqli = new mysqli("localhost", $username, $password, $database);
$query="CREATE TABLE tablename(id int(6) NOT NULL auto_increment,first varchar(15) NOT NULL,last varchar(15) NOT NULL,field1-name varchar(20) NOT NULL,field2-name varchar(20)NOT NULL,field3-name varchar(20) NOT NULL,field4-name varchar(30) NOT NULL, field5-name varchar(30)NOT NULL,PRIMARY KEY (id),UNIQUE id (id),KEY id_2 (id))";
$mysqli->query("$query");
$mysqli->close();
?>
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
```

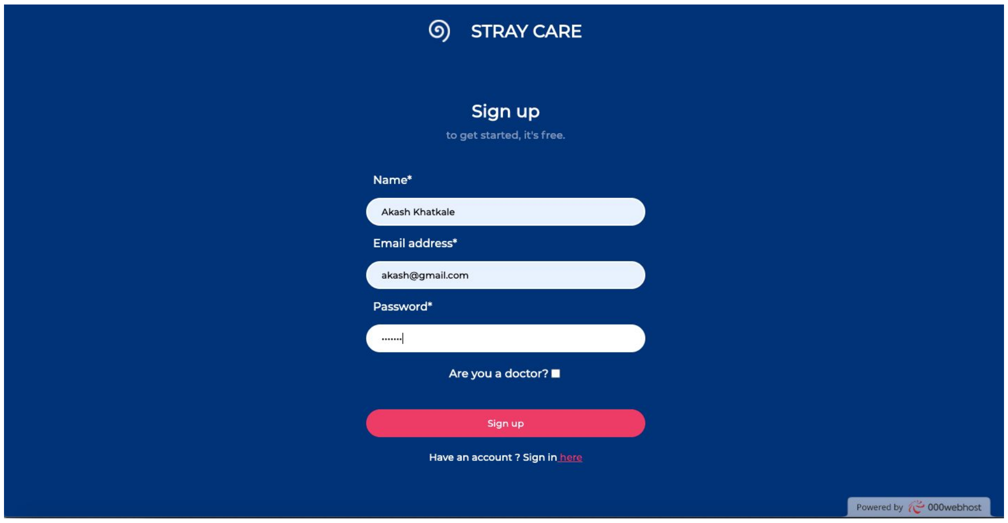
```
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
   die("Connection failed: " . mysqli_connect_error());
}
$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
   // output data of each row
   while($row = mysqli_fetch_assoc($result)) {
      echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"]. "<br>";
   }
} else {
   echo "0 results";
}
mysqli_close($conn);
?>
```

Code(insert):



```php
<?php
session_start();
if(isset($_POST["Submit"])){
    $name = $_POST["Name"];
    $email = $_POST["Email"];
    $pass = $_POST["Password"];
    if(!empty($name) || !empty($email) || !empty($pass)){
        if(!preg_match("[illegible]", $email)){
            header("location:signup.php?msg=Enter valid email address");
        }else{
            if(strlen($pass)<6){
                header("location:signup.php?msg=Password should be greater than 6 characters");
            }else{
                $host = "localhost";
                $username = "id[illegible]_user";
                $password = "[illegible]";
                $dbname = "id[illegible]_straycare";
                $conn = mysqli_connect($host,$username, $password, $dbname);

                if(isset($_POST["Doctor"])){
                    $query = "INSERT INTO doctors(name,email,password) VALUES('$name', '$email', '$pass')" ;
                }else{
                    $query = "INSERT INTO users(name,email,password) VALUES('$name', '$email', '$pass')" ;
                }
                $result = mysqli_query($conn, $query);

                if($result){
                    $_SESSION["name"] = $name;
                    $_SESSION["email"] = $email;
                    header("location:signup.php?msg=An error occurred");
                }else{
                    header("location:signup.php?msg=An error occurred");
                }
                mysqli_close($conn);
            }
        }

    }else{
        header("location:signup.php?msg=All fields are required");
    }
}else{
    header("location:signup.php");
}
```

Output:



Sign up

to get started, it's free.

Name*

Akash Khatkale

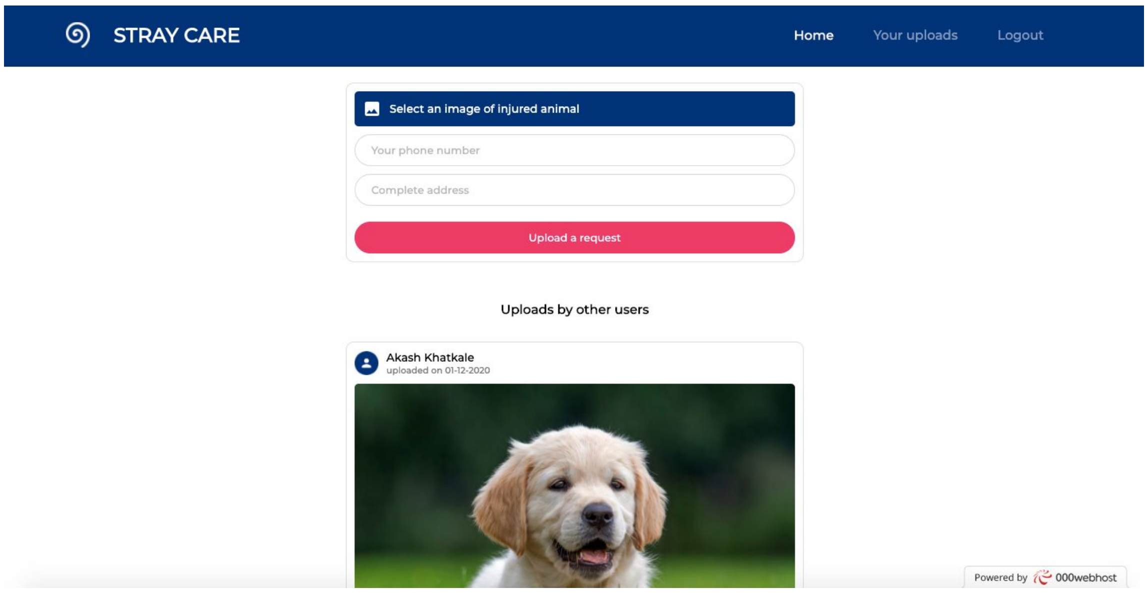
Email address*

akash@gmail.com

Password*

•••••••

Are you a doctor? ■


STRAY CARE

Home Your uploads Logout

Select an image of injured animal

Your phone number

Complete address

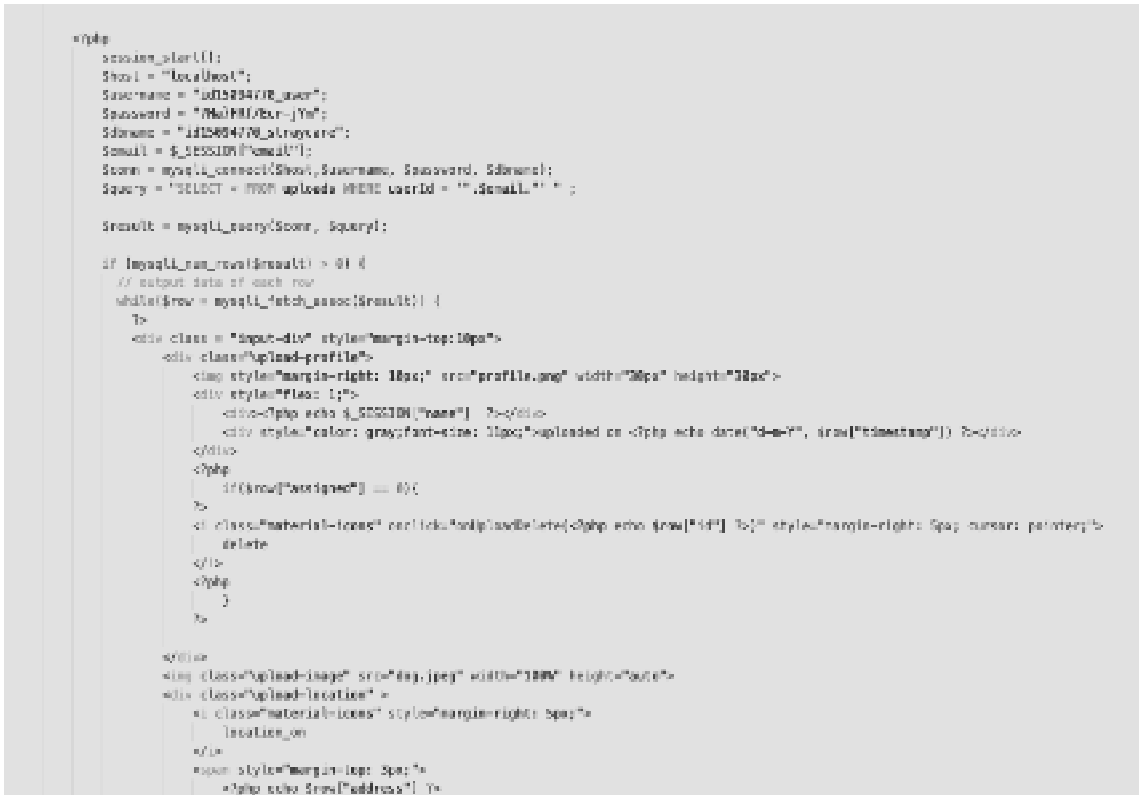
Upload a request

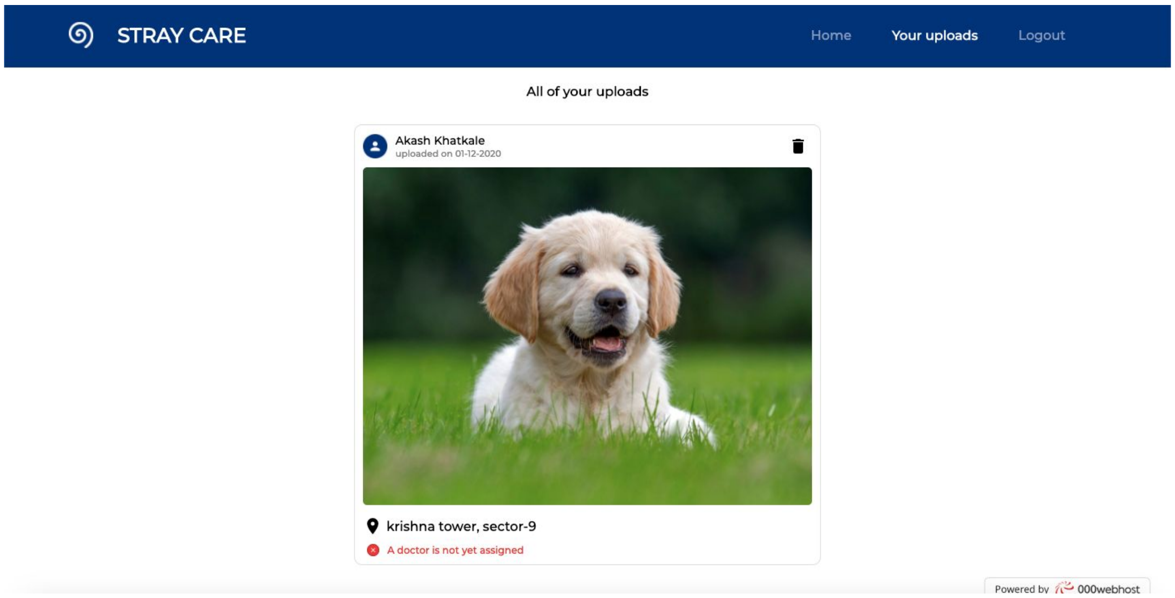
Uploads by other users

Akash Khatkale
uploaded on 01-12-2020

Powered by 000webhost

Code(read);


```
<?php
    session_start();
    $host = "localhost";
    $username = "id15894770_user";
    $password = "/MaJ?RI/bxr-jYm";
    $dbname = "id15894770_straycare";
    $email = $_SESSION["email"];
    $conn = mysqli_connect($host,$username, $password, $dbname);
    $query = "SELECT * FROM uploads WHERE userId = '".$email."' " ;

    $result = mysqli_query($conn, $query);

    if (mysqli_num_rows($result) > 0) {
      // output data of each row
      while($row = mysqli_fetch_assoc($result)) {
        ?>
        <div class = "input-div" style="margin-top:10px">
            <div class="upload-profile">
                <img style="margin-right: 10px;" src="profile.png" width="30px" height="30px">
                <div style="flex: 1;">
                    <div><?php echo $_SESSION["name"]  ?></div>
                    <div style="color: gray;font-size: 11px;">uploaded on <?php echo date("d-m-Y", $row["timestamp"]) ?></div>
                </div>
                <?php
                    if($row["assigned"] == 0){
                ?>
                <i class="material-icons" onclick="onUploadDelete(<?php echo $row["id"] ?>)" style="margin-right: 5px; cursor: pointer;">
                    delete
                </i>
                <?php
                    }
                ?>

            </div>
            <img class="upload-image" src="dog.jpeg" width="100%" height="auto">
            <div class="upload-location" >
                <i class="material-icons" style="margin-right: 5px;">
                    location_on
                </i>
                <span style="margin-top: 5px;">
                    <?php echo $row["address"] ?>
```



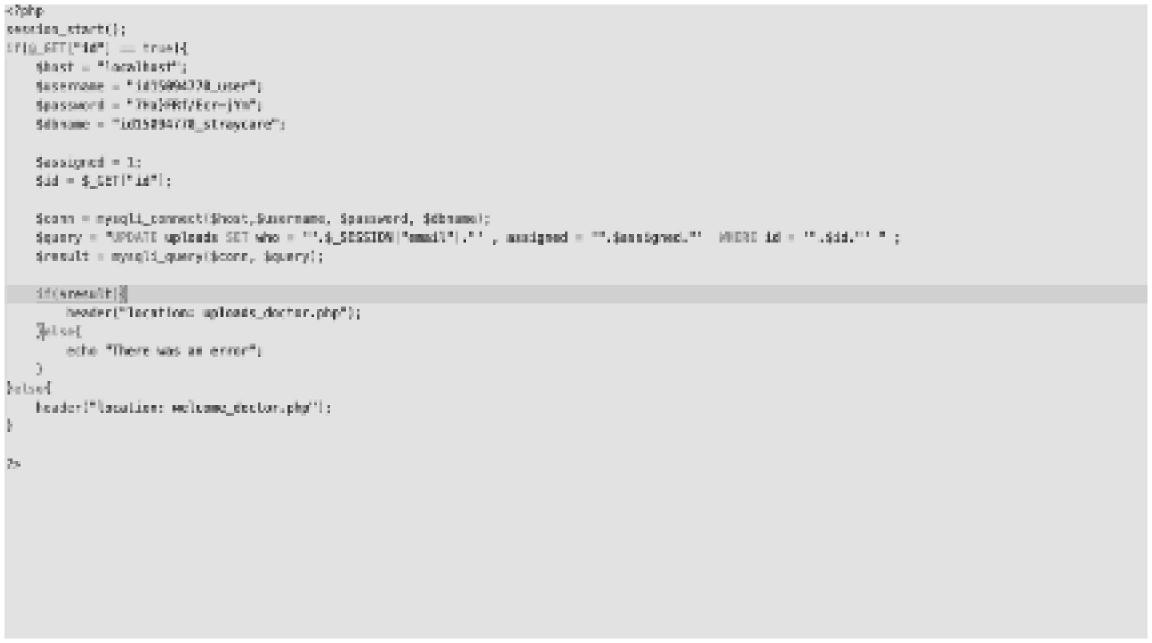
All of your uploads

Akash Khatkale
uploaded on 01-12-2020

krishna tower, sector-9

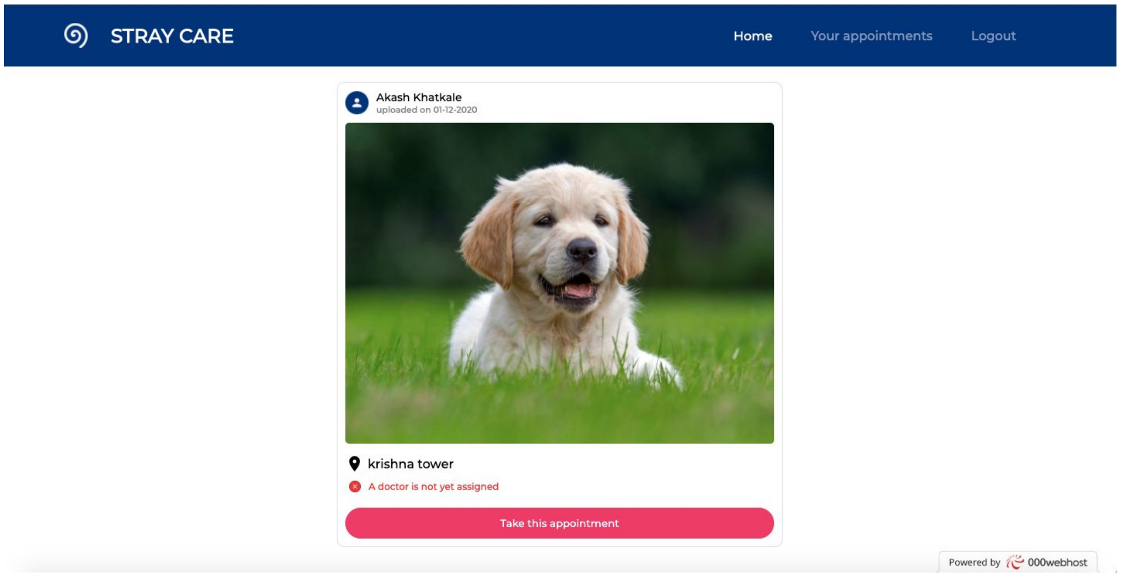
A doctor is not yet assigned

Code(update):

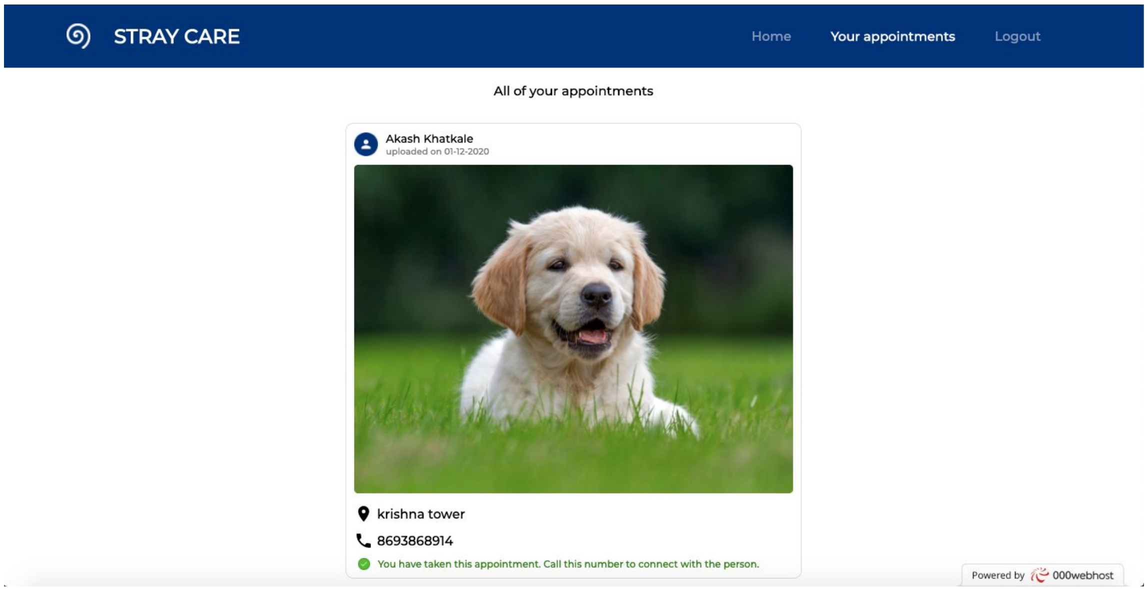


```
<?php
session_start();
if($_GET["id"] == true){
    $host = "localhost";
    $username = "id15004770_user";
    $password = "7ha}PRf/Ecr-jYh";
    $dbname = "id15004770_straycare";

    $assigned = 1;
    $id = $_GET["id"];

    $conn = mysqli_connect($host,$username, $password, $dbname);
    $query = "UPDATE uploads SET who = '".$_SESSION["email"]."' , assigned = '".$assigned."'  WHERE id = '".$id."' " ;
    $result = mysqli_query($conn, $query);

    if($result){
        header("location: uploads_doctor.php");
    }else{
        echo "There was an error";
    }
}else{
    header("location: welcome_doctor.php");
}

?>
```

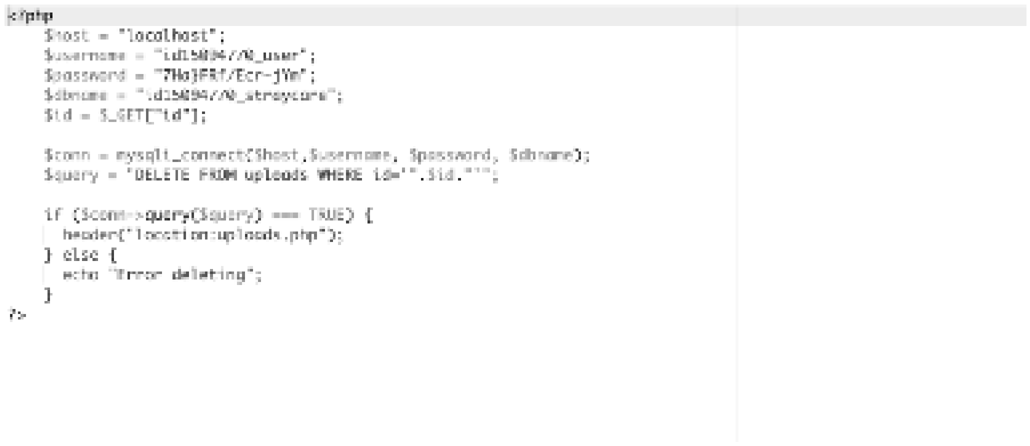
Output(before clicking the pink button):



After clicking the pink button

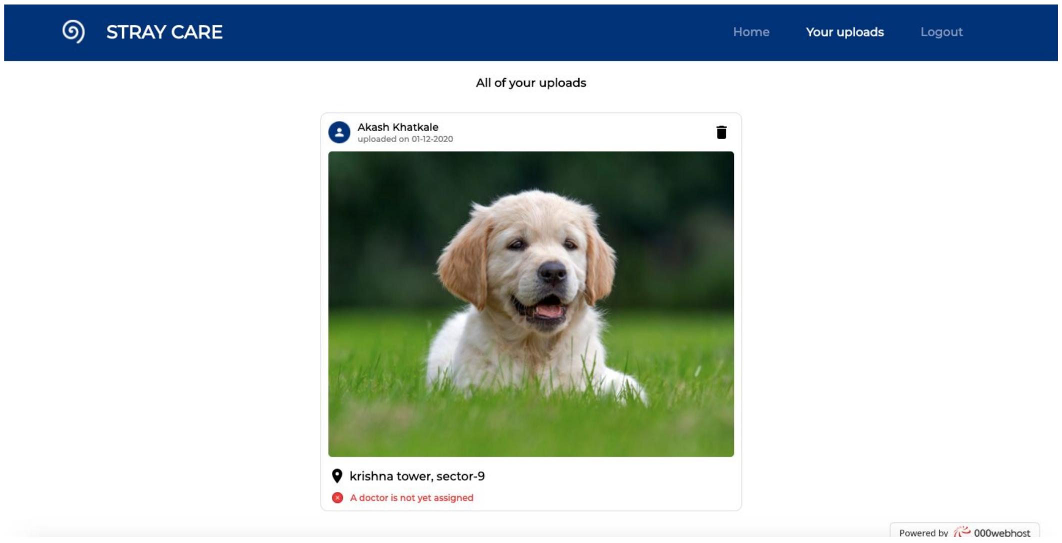


Code(delete);



```php
<?php
    $host = "localhost";
    $username = "id15094770_user";
    $password = "7Ho}FRf/Ecr-jYm";
    $dbname = "id15094770_straycare";
    $id = $_GET["id"];

    $conn = mysqli_connect($host,$username, $password, $dbname);
    $query = "DELETE FROM uploads WHERE id='".$id."'";

    if ($conn->query($query) === TRUE) {
      header("location:uploads.php");
    } else {
      echo "Error deleting";
    }
?>
```

Output:

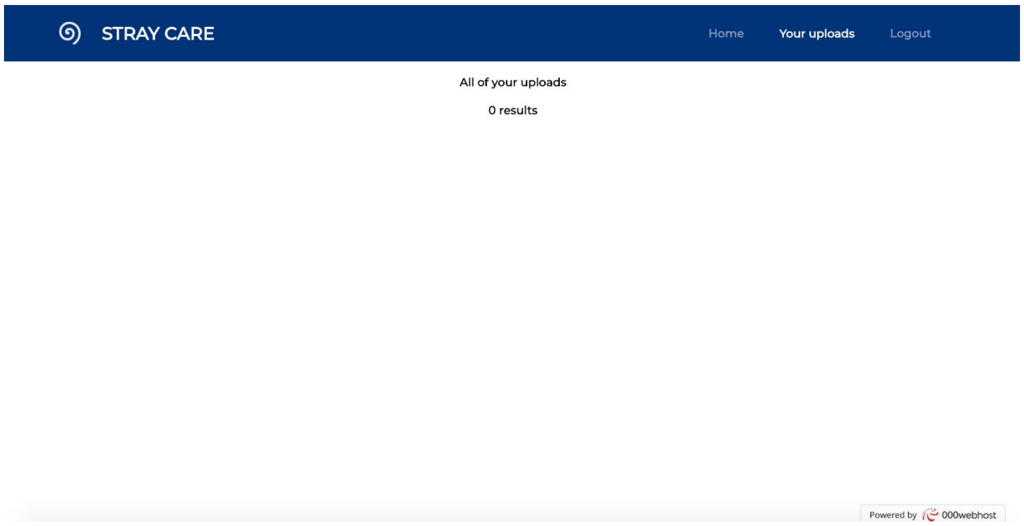


All of your uploads

Akash Khatkale
uploaded on 01-12-2020

krishna tower, sector-9

A doctor is not yet assigned



All of your uploads

0 results

Chapter 13

RIA using AJAX

JAX stands for Asynchronous JavaScript and XML. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and Java Script. Ajax uses XHTML for content, CSS for presentation, along with Document Object Model and JavaScript for dynamic content display. Conventional web applications transmit information to and from the sever using synchronous requests. It means you fill out a form, hit submit, and get directed to a new page with new information from the server. With AJAX, when you hit submit, JavaScript will make a request to the server, interpret the results, and update the current screen. In the purest sense, the user would never know that anything was even transmitted to the server.

XML is commonly used as the format for receiving server data, although any format, including plain text, can be used. AJAX is a web browser technology independent of web server software. A user can continue to use the application while the client program requests information from the server in the background. Intuitive and natural user interaction. Clicking is not required, mouse movement is a sufficient event trigger. Data-driven as opposed to page-driven.

Rich Internet Application Technology

AJAX is the most viable Rich Internet Application (RIA) technology so far. It is getting tremendous industry momentum and several tool kit and frameworks are emerging. But at the same time, AJAX has browser incompatibility and it is supported by JavaScript, which is hard to maintain and debug. AJAX is Based on Open Standards

AJAX is based on the following open standards −

- Browser-based presentation using HTML and Cascading Style Sheets (CSS).
- Data is stored in XML format and fetched from the server.
- Behind-the-scenes data fetches using XMLHttpRequest objects in the browser.
- JavaScript to make everything happen.

The XMLHttpRequest Object

All modern browsers support the XMLHttpRequest object. The XMLHttpRequest object can be used to exchange data with a server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

Send a Request To a Server

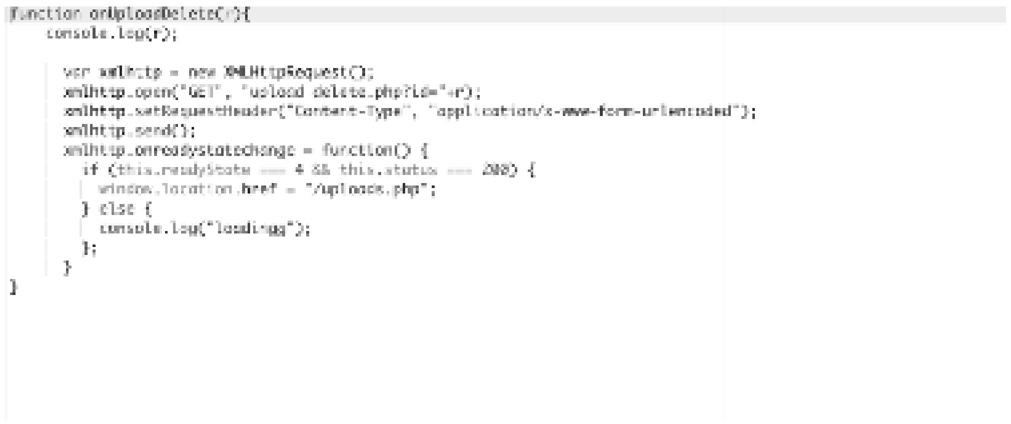
To send a request to a server, we use the open() and send() methods of the XMLHttpRequest object:

```
xhttp.open("GET", "ajax_info.txt", true);

xhttp.send();
```

The onreadystatechange Property The readyState property holds the status of the XMLHttpRequest. The onreadystatechange property defines a function to be executed when the readyState changes. The status property and the statusText property holds the status of the XMLHttpRequest object.

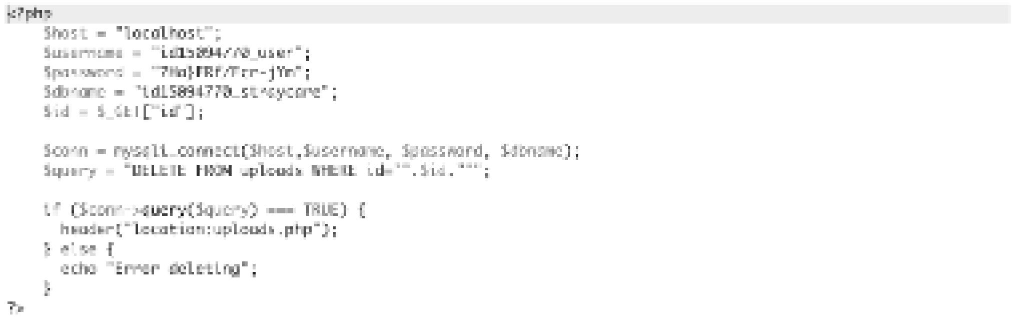
```
function loadDoc() {
   var xhttp = new XMLHttpRequest();
   xhttp.onreadystatechange = function() {
      if (this.readyState == 4 && this.status == 200) {
         document.getElementById("demo").innerHTML =
         this.responseText;
      }
   };
   xhttp.open("GET", "ajax_info.txt", true);
   xhttp.send();
}
```

RIA CODE



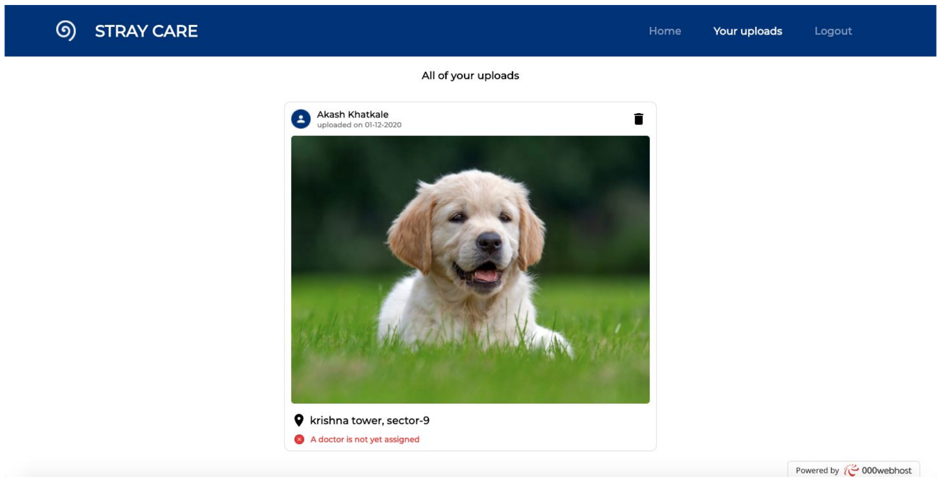
```
function onUploadDelete(r){
    console.log(r);

      var xmlhttp = new XMLHttpRequest();
      xmlhttp.open("GET", "upload_delete.php?id="+r);
      xmlhttp.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
      xmlhttp.send();
      xmlhttp.onreadystatechange = function() {
        if (this.readyState === 4 && this.status === 200) {
          window.location.href = "/uploads.php";
        } else {
          console.log("loadingg");
        };
      }
}
```

upload_delete.php

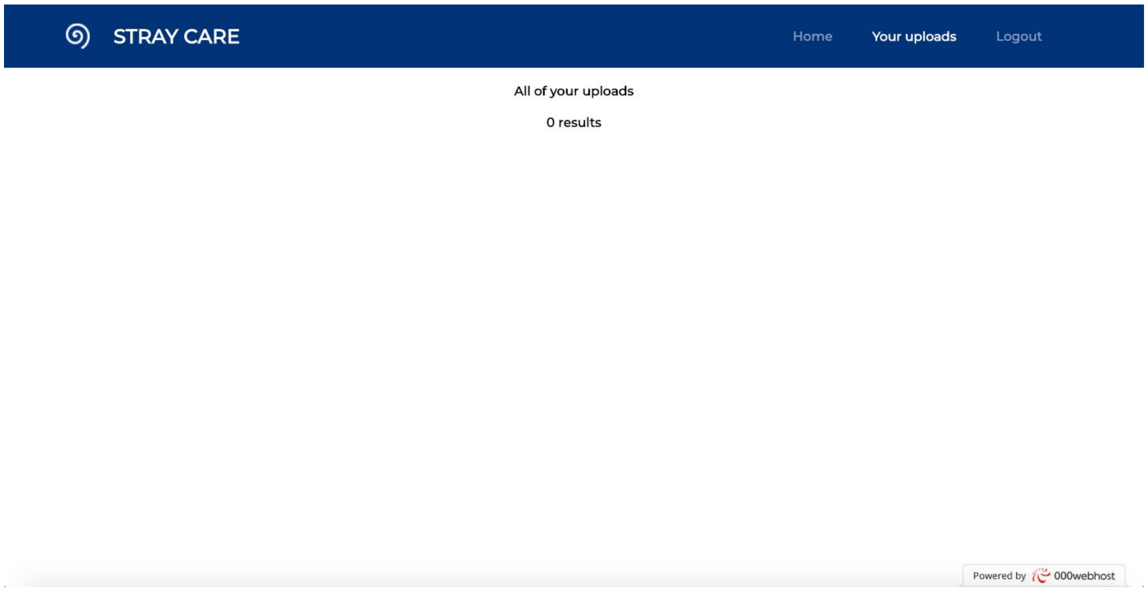


```
<?php
    $host = "localhost";
    $username = "id15894770_user";
    $password = "7Ha}FRf/Trr-jYn";
    $dbname = "id15894770_straycare";
    $id = $_GET["id"];

    $conn = mysqli_connect($host,$username, $password, $dbname);
    $query = "DELETE FROM uploads WHERE id='".$id."'";

    if ($conn->query($query) === TRUE) {
      header("location:uploads.php");
    } else {
      echo "Error deleting";
    }
?>
```

Before clicking the delete button



After clicking the delete button



All of your uploads

0 results

Chapter 14
Web hosting

Web hosting is a service that allows organizations and individuals to post a website or web page onto the Internet. A web host, or web hosting service provider, is a business that provides the technologies and services needed for the website or webpage to be viewed in the Internet. Websites are hosted, or stored, on special computers called servers. When Internet users want to view your website, all they need to do is type your website address or domain into their browser. Their computer will then connect to your server and your web pages will be delivered to them through the browser.

Most hosting companies require that you own your domain in order to host with them. If you do not have a domain, the hosting companies will help you purchase one.

Here are some features you should be expecting from your hosting provider:

Email Accounts : As mentioned earlier, most hosting providers require users to have their own domain name. With a domain name (e.g. www.yourwebsite.com) and email account features provided by your hosting company, you can create domain email accounts (e.g. yourname@yourwebsite.com).

FTP Access : The use of FTP lets you upload files from your local computer to your web server. If you build your website using your own HTML files, you can transfer the files from your computer to the web server through FTP, allowing your website to be accessed through the internet.

Here is our website URL :

https://hawkish-pain.000webhostapp.com/

Normal user : https://hawkish-pain.000webhostapp.com/login.php

Email : user@gmail.com

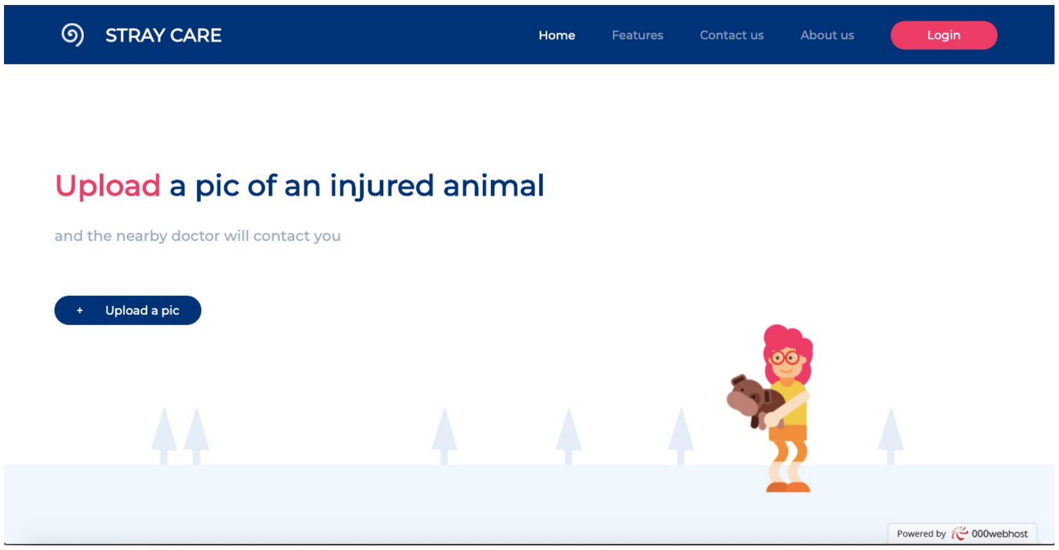
Password : 1234567

Admin user : https://hawkish-pain.000webhostapp.com/login.php

Email : admin@gmail.com

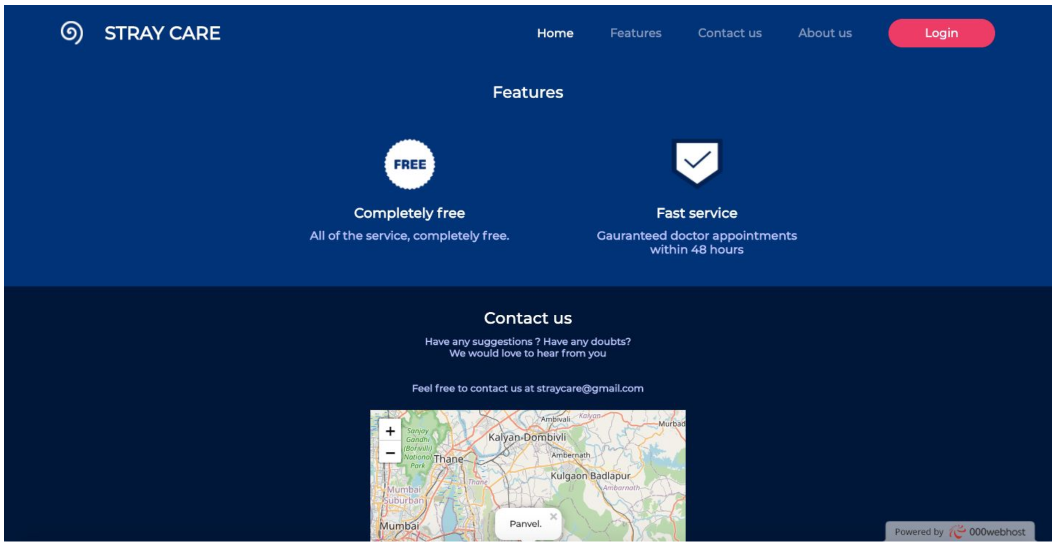
Password : 1234567

Snapshot of Landing page



Upload a pic of an injured animal

and the nearby doctor will contact you



Features

Completely free

All of the service, completely free.

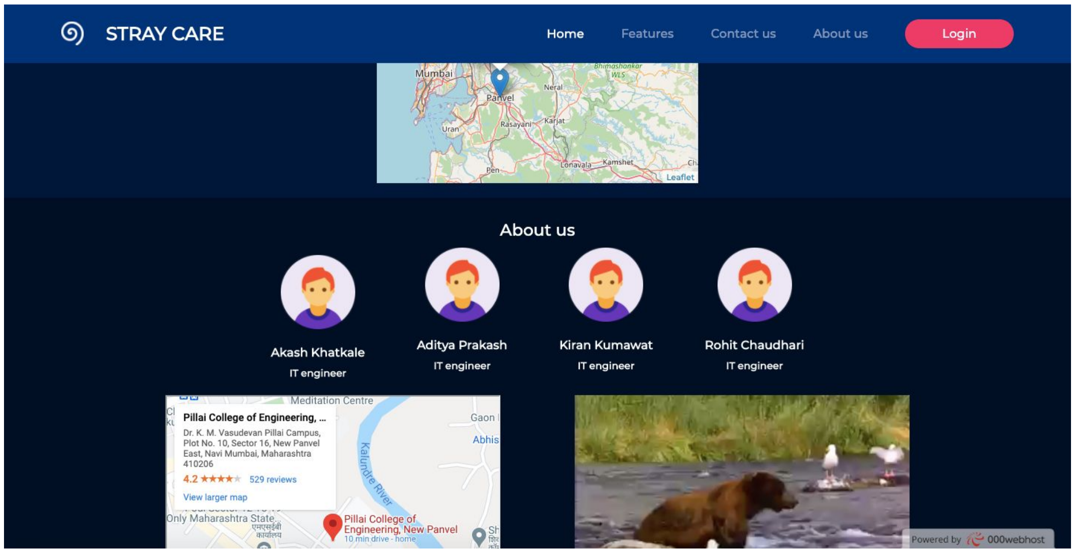
Fast service

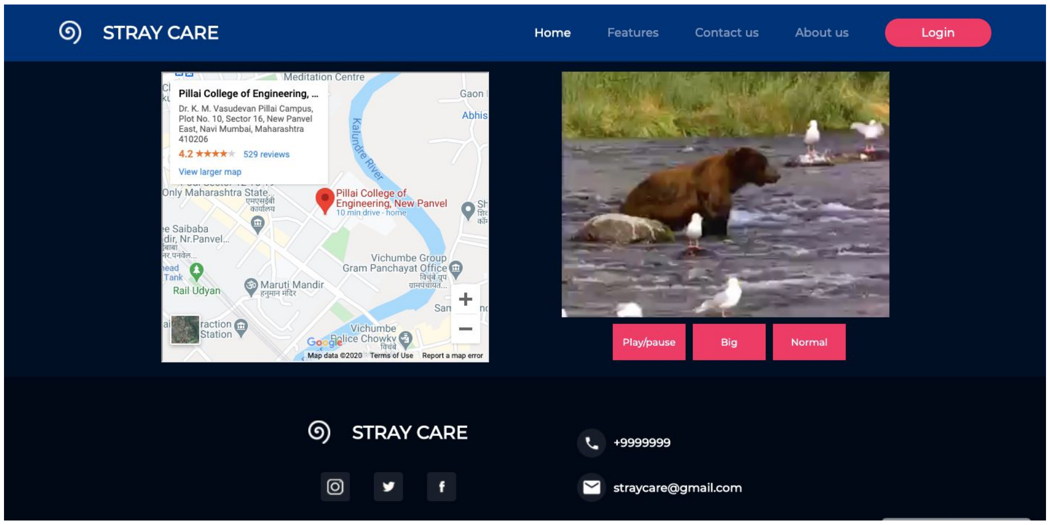
Gauranteed doctor appointments within 48 hours

Contact us

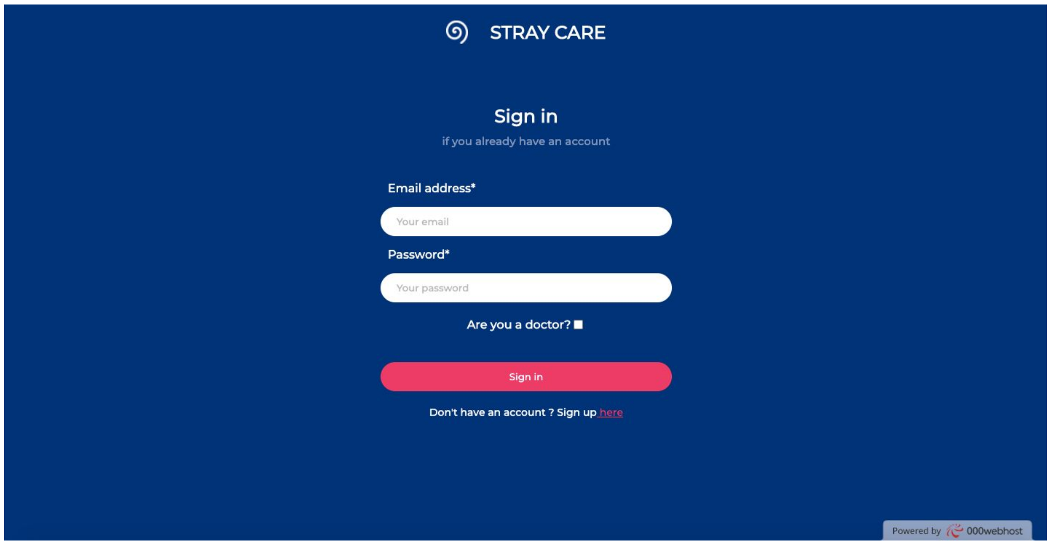
Have any suggestions ? Have any doubts?
We would love to hear from you

Feel free to contact us at straycare@gmail.com

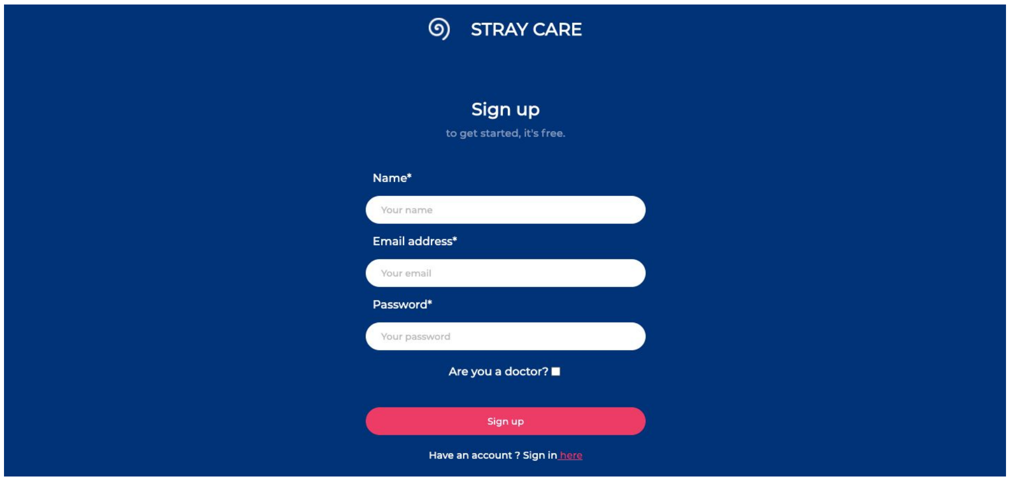




Snapshot of Login Page



Snapshot of Registration Page


STRAY CARE

Sign up

to get started, it's free.

Name*

Your name

Email address*

Your email

Password*

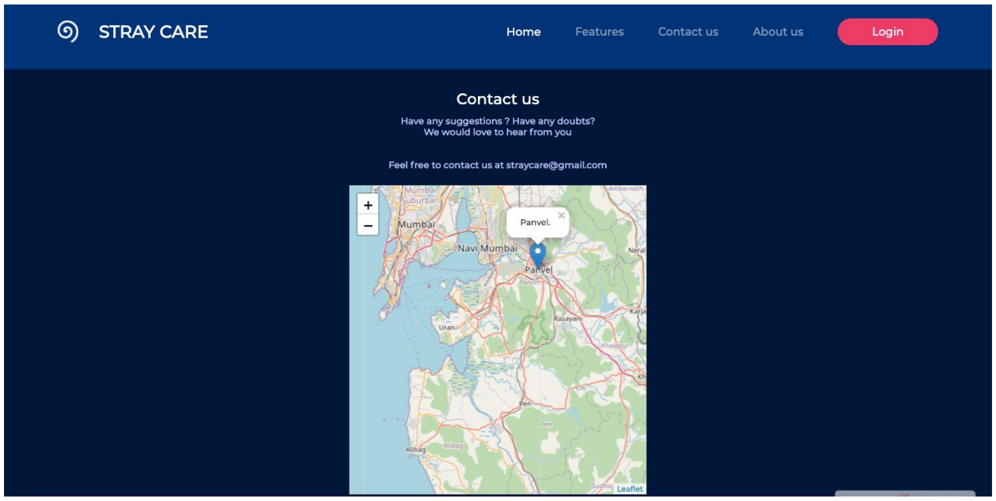
Your password

Are you a doctor?

Sign up

Have an account ? Sign in here

Snapshot of Contact us page


STRAY CARE

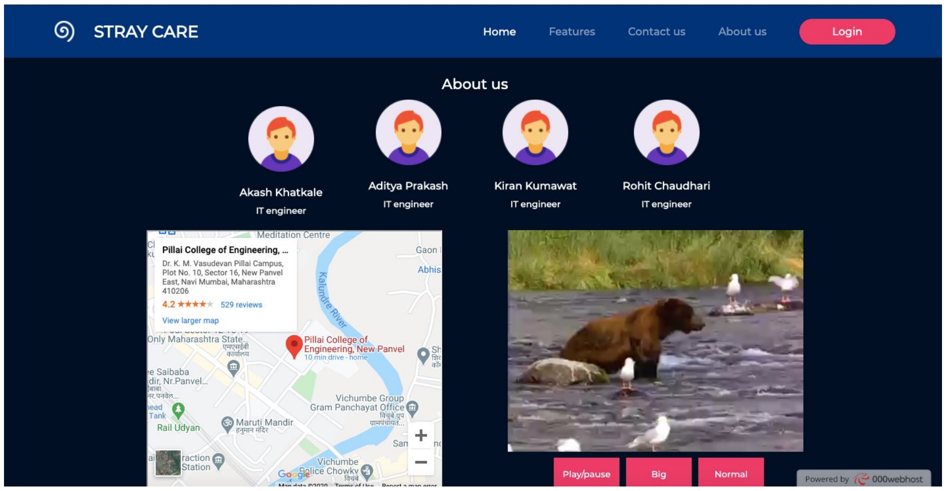
Home Features Contact us About us Login

Contact us

Have any suggestions ? Have any doubts?
We would love to hear from you

Feel free to contact us at straycare@gmail.com

Snapshot of About us page


STRAY CARE

Home Features Contact us About us Login

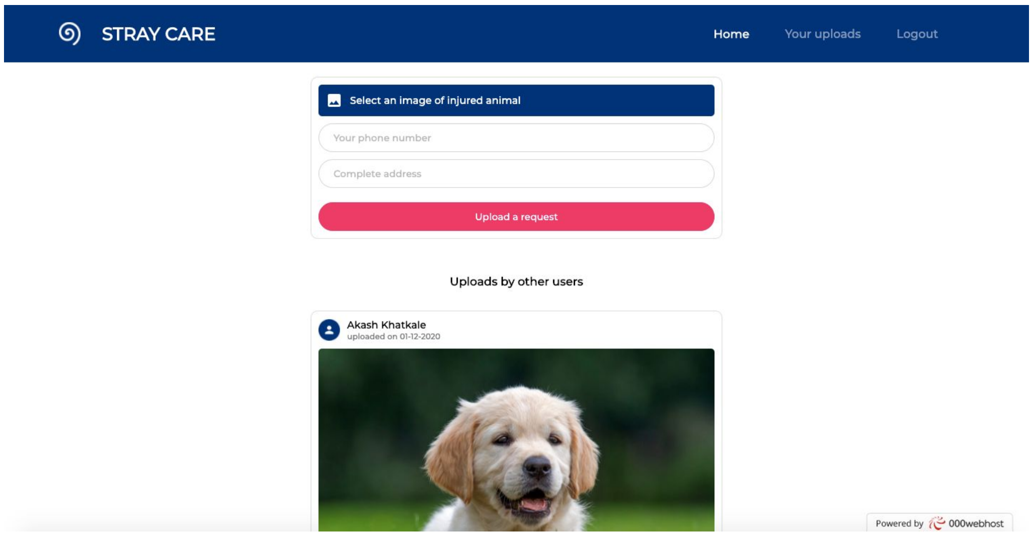
About us

Akash Khatkale	Aditya Prakash	Kiran Kumawat	Rohit Chaudhari
IT engineer	IT engineer	IT engineer	IT engineer

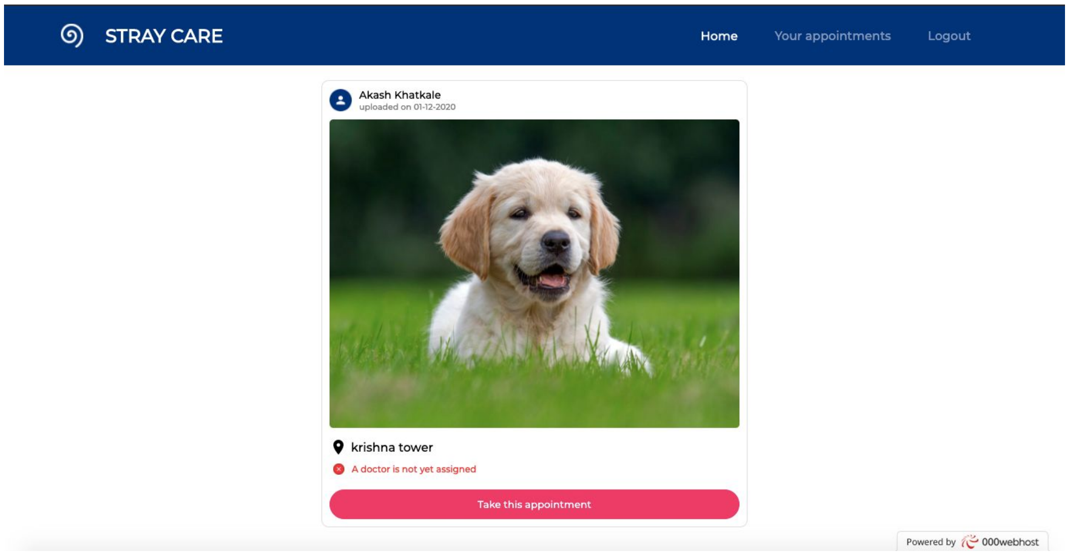
Pillai College of Engineering, ...
Dr. K. M. Vasudevan Pillai Campus, Plot No. 10, Sector 16, New Panvel East, Navi Mumbai, Maharashtra 410206
4.2 ★★★★ 529 reviews
View larger map

Pillai College of Engineering, New Panvel
Kalundre River
Vichumbe Group Gram Panchayat Office
Maruti Mandir
Rail Udyan
Play/pause
Big
Normal
Powered by 000webhost

Snapshot of User Home page



Snapshot of Admin Home page



Acknowledgement

We are thankful to everyone who provided us the opportunity to complete this report. We give special gratitude to our Principal Dr. Sandeep Joshi, who always encourages us to do innovative things that will help us to increase our knowledge and serve the society in one or the other way.

We would also like to thank our H.O.D of the information technology Dr. Satish Kumar Verma for this opportunity and for providing us with enough resources to complete our tasks.

We would also like to thank our Prof. Dhiraj Amin who always motivates us and makes our concepts clear and also helps us to solve our doubts and perform well in our academics.

Akash Khatkale
Aditya Chattikal
Kiran Kumawat
Rohit Chaudhari