
VEHICLE DETECTOR

| SEARCH MADE EASY |

Batch : D14

Team members:

Kumaran KM (211418104132)

Pradeish C (211418104196)

Naveen SR (211418104172)



VEHICLE DETECTOR

SEARCH MADE EASY

Abstract:

- It is an application that helps the owners to track their vehicle when they get lost by using vehicle's unique information.
- Our main goal is to provide the owners with a searching facility across the city using a Public Camera GPS, and mobile capturing technique without raising the complaint to the cop.
- The tracking system is done by using captured images by unknown users when they have captured during suspicious activity of that vehicle.



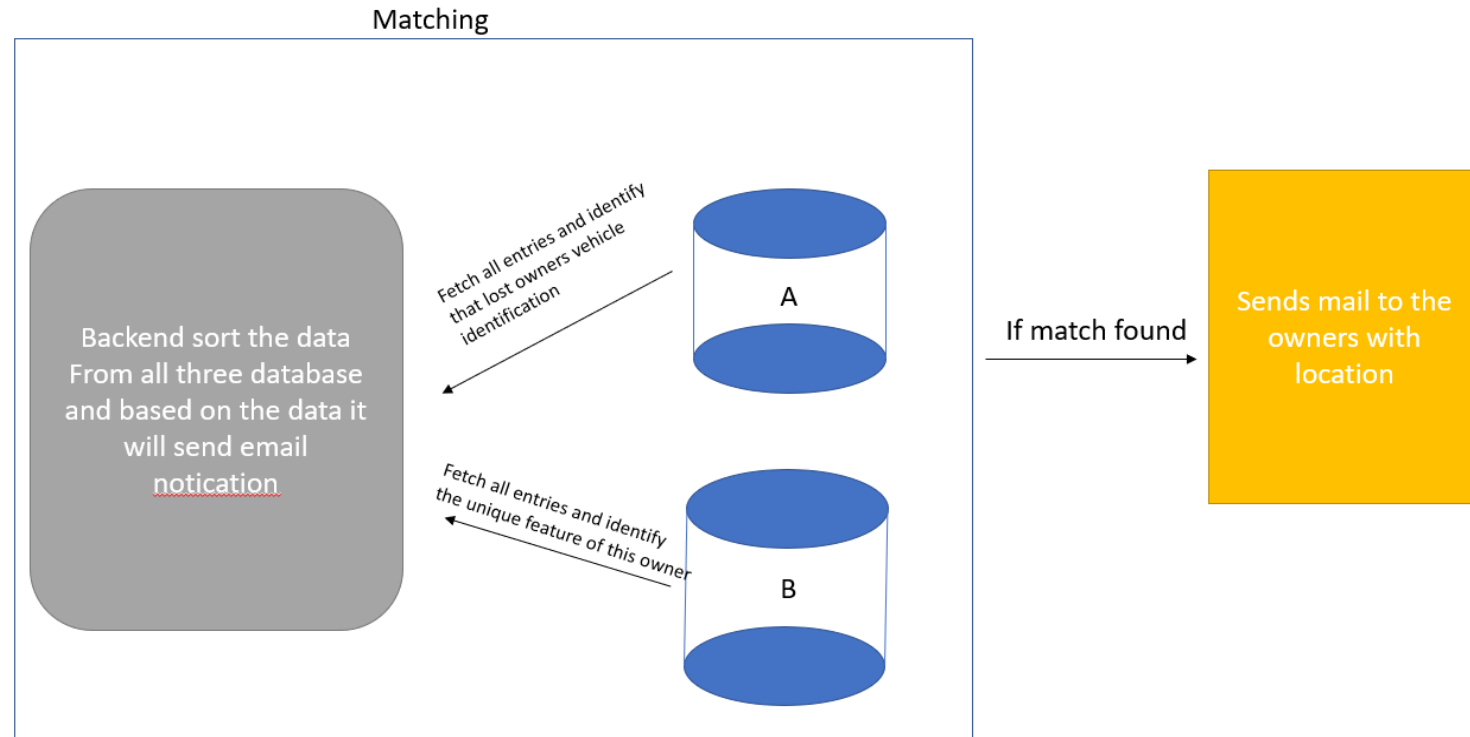
Technical Stack:

- Language : Python
- Library : Keras, Tensorflow
- Database : Excel sheet
- Tool : Vs code
- Deploy : GitHub Pages



System Architecture

Search for vehicle option



Implementation of the Algorithms / Coding

Step 1: Information are collected by the owners with the help of form and it get stored on registration database

Step 2: Similarly data are collected by the helpers with the help of google form and it get stored on helper database

Step 3: If vehicle gets lost, the owners can ping on our platform and based on the profile the information are collected

Step 4: On the other hand, the collected image from the helpers get processed and by using image recognition technique we will sort the image with some properties



Step 5: The sorted images are updated on helper database with properties like color, license plate, seat color, vehicle type, etc.

Step 6: So we got enough data and it get processed. Now we will move for backend part

Step 7: On the backend, our sever code traverse to all the three database(Register, Helper, Lost) and it will find the matches and common details between these databases

Step 8: Once we have found matches on these database, the email notification with location and captures image will be sent to owners who have lost their vehicle



Implementation

Data that are collected at the time of registration

```
<form >  
  <input type="text" placeholder="Full Name" >  
  <input type="mail" placeholder="Email Id" >  
  <input type="number" placeholder="Phone Number" >  
  <input type="text" placeholder="Register number" >  
  <input type="text" placeholder="Brand name" >  
  <input type="text" placeholder="Vehicle colour" >  
  <input type="text" placeholder="Other Information" >  
</form>
```



Data that are collected when user pings for vehicle lost

```
<form>  
  <input type="email" placeholder="Enter your Email Address" >  
  <input type="date" required onclick="datee()">  
  <button type="submit" onclick="fetchInfo()">Find</button>  
</form>
```

Data that are collected when helper post the images they captured

```
<form>  
  <input type="images" placeholder="Upload your image" >  
  <input type="text" " placeholder="Enter the location of the image">  
  <button type="submit" >Upload</button>  
</form>
```

Code for color detection using python

```
def getRGB(im):  
    im = cv2.resize(im, (150,150)) #im.resize((150, 150))  
    shape = ar.shape  
    ar = ar.reshape(scipy.product(shape[:2]), shape[2]).astype(float)  
    codes, dist = scipy.cluster.vq.kmeans(ar, NUM_CLUSTERS)  
    vecs, dist = scipy.cluster.vq.vq(ar, codes)  
    counts, bins = scipy.histogram(vecs, len(codes))  
    index_max = scipy.argmax(counts)  
    peak = codes[index_max]  
    r,g,b = (int(c) for c in peak)  
    return r,g,b
```

Code for traversing the database and finding the match

```
for b in entries2:
    for a in entries1:
        if b[0] == a[0]: # matching mail
            matchess = []
            linkk = []
            for c in entries3:

                if c[1].casefold() == a[1].casefold(): # matching color
                    # storing location in list
                    data_matched = [c[0]]
                    matchess.append(data_matched)
                    # storing image links in list
                    link_matched = [c[2]]
                    linkk.append(link_matched)
            continue
```

Testing / Performance Analysis

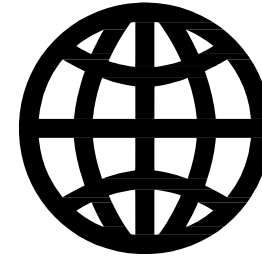
S. No	Test Case	Test Steps	Test Results
1	Verify Login functionalities with the correct condition	Enter mail id: abc@sample.com Enter password: xyz	Redirect to owners page
2	Verify Login functionalities with the wrong condition	Enter mail id: abc@sample.com Enter password: Empty	Password not found stay in the same owner page

S. No	Test Case	Test Steps	Test Results
3	Sending Images to the platform	Send photos here: sample.jpg Send Location: Poonamale, Chennai	Redirect to helper page
4	Sending Images to the platform without proper data	Send photos here: sample.jpg Send Location: Empty	Location not found. Staying the same page

Requirement Analysis:

Hardware:

- Desktop or Mobile
- Internet Connection

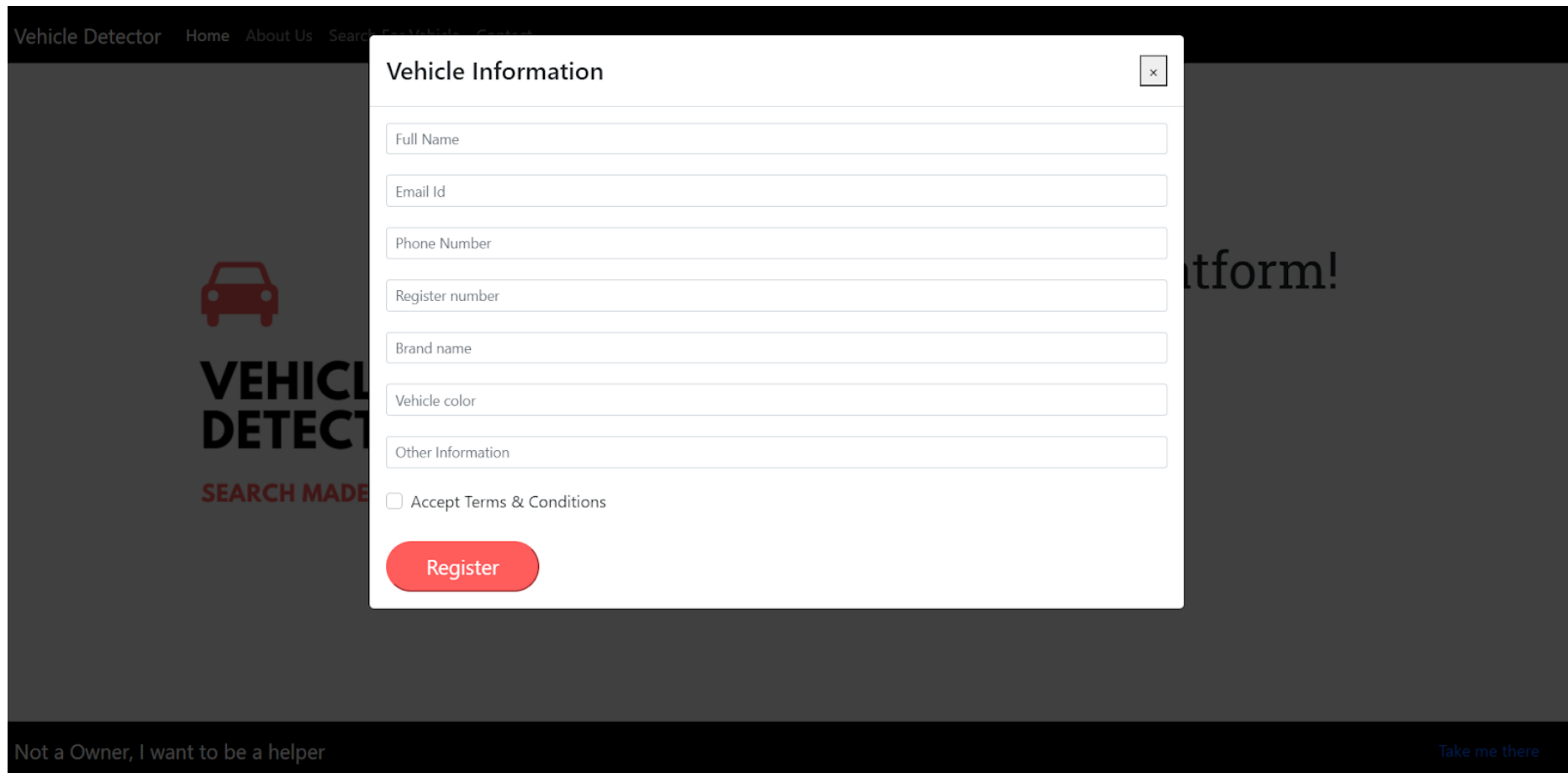


Software:

- Any advance browser
- And any operating system



Output window



The screenshot displays a web application interface. At the top, a dark navigation bar contains the text "Vehicle Detector" followed by links: "Home", "About Us", "Search For Vehicle", and "Contact". The main background is dark grey with a red car icon on the left and the text "VEHICLE DETECTOR" and "SEARCH MADE" in large, bold letters. A white modal window titled "Vehicle Information" is centered, featuring a close button (X) in the top right corner. The form inside the modal includes the following fields: "Full Name", "Email Id", "Phone Number", "Register number", "Brand name", "Vehicle color", and "Other Information". Below these fields is a checkbox labeled "Accept Terms & Conditions". At the bottom of the modal is a red "Register" button. In the bottom left corner of the page, there is a link that says "Not a Owner, I want to be a helper". In the bottom right corner, there is a link that says "Take me there".

Vehicle Detector Home About Us Search For Vehicle Contact

Vehicle Information

Full Name

Email Id

Phone Number

Register number

Brand name

Vehicle color

Other Information

☐ Accept Terms & Conditions

Register

Not a Owner, I want to be a helper

Take me there

Search For Vehicle



Vehicle got lost?

Enter your Email Address

When You Lost?

dd-mm-yyyy



Find

Not a Owner, I want to be a helper

[Take me there](#)

Vehicle Matching Inbox x



Vehicle Detector Application <vehicledetectorproject@gmail.com>
to me ▾

10:00 PM (0 minutes ago) ☆ ↶ ⋮

Hello Kumaran KM!

We have collected certain images based on your features and thier locations are:

- 1) Delhi and the image link is <https://drive.google.com/open?id=1mYVKLVCqrUCmgsAi2kv-sQIDEOSVpDkK>
- 2) Bangalore and the image link is <https://drive.google.com/open?id=1K7LgxtqztD21AjJNLwl2qLNgDWdTe8pp>

2 Attachments



↶ Reply

➡ Forward

Conclusion & Future Enhancements

The above-proposed system has provided comprehensive methods and an overview of the application that detects vehicles with the helping image capturing technique.

If there are any server issues successes or fraudulent information, then this whole system is taken care of by the admin. Hence this platform made the owners find the vehicle without any police complaints.



-
- First, we are going to import mongo as a cloud database(Atlas) for this project so that we can add or edit enormous amounts of data easily and quickly.
 - Then we would like to do web hosting via a digital ocean network instead of using GitHub pages and we would like to buy the domain for our project
 - Then we would like to feed more datasets into our machine learning framework and add additional features like seat identifying, car steering color, scratch features, etc for detecting a vehicle.





VEHICLE DETECTOR

SEARCH MADE EASY

Project Url:

<https://projectdetector.github.io/FrontEndCode/>

Email us:

vehicledetectorproject@gmail.com

