MINI PROJECT

(2021-2022)

“Object Detection”

Project Report

Institute of Engineering & Technology

Submitted By -

Surya Raj (191500832)

Priya Tomar (191500603)

Under the Supervision Of

Mr. Mandeep Singh

Technical Trainer

Department of Computer Engineering & Applications

Department of Computer Engineering and Applications

GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,

Chaumuha, Mathura – 281406 U.P (India)

Declaration

I/we hereby declare that the work which is being presented in the Bachelor of Technology, project **"Object Detection"**, in partial fulfillment of the requirements for the award of the ***Bachelor of Technology*** in Computer Science and Engineering and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of my/our own work carried under the supervision of **Mr. Mandeep Singh, Technical Trainer, Dept. of CEA, GLA University.**

The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

Sign: *Surya Raj*	**Sign**: *Priya Tomar*
Name of Candidate: Surya Raj	**Name of Candidate**: Priya Tomar
University Roll No.:191500832	**University Roll No.**:191500603

Department of Computer Engineering and Applications

GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,

Chaumuha, Mathura – 281406 U.P (India)

Certificate

This is to certify that the project entitled "Object Detection", carried out in Mini Project – I Lab, is a Bona Fide work by Surya Raj and Priya Tomar, which is submitted in partial fulfillment of the requirements for the award of the degree Bachelor of Technology (Computer Science & Engineering).

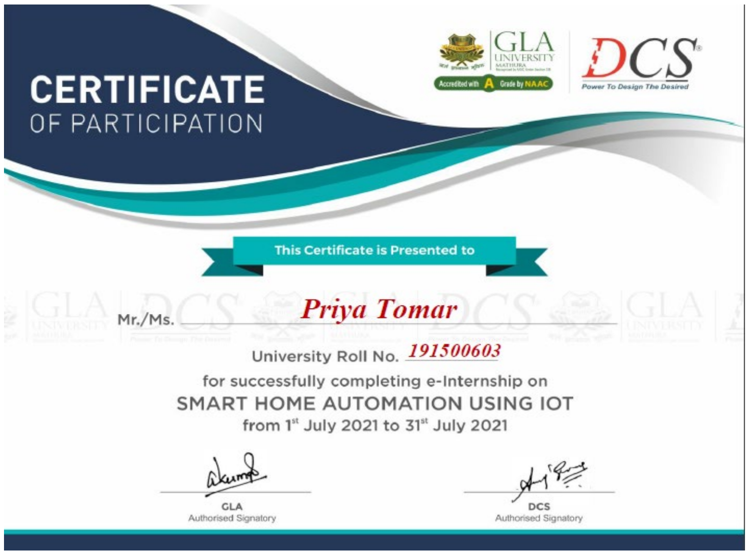
Signature of Supervisor:

Name of Supervisor: Mr. Mandeep Singh

Date:

Training Certificates

Priya Tomar :



GLA UNIVERSITY MATHURA

Accredited with A Grade by NAAC

DCS

Power To Design The Desired

CERTIFICATE

OF PARTICIPATION

This Certificate is Presented to

Mr./Ms. *Priya Tomar*

University Roll No. *191500603*

for successfully completing e-Internship on

SMART HOME AUTOMATION USING IOT

from 1st July 2021 to 31st July 2021

[illegible]

GLA

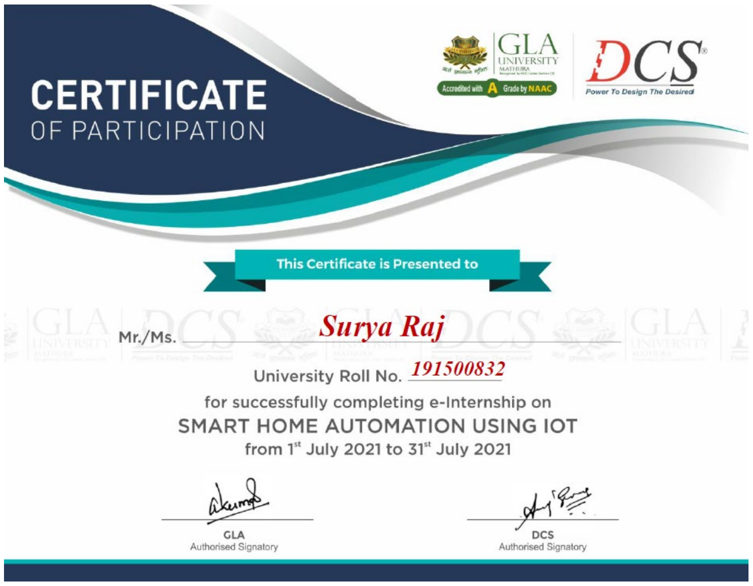
Authorised Signatory

[illegible]

DCS

Authorised Signatory

Surya Raj:



GLA UNIVERSITY MATHURA

Accredited with A Grade by NAAC

DCS

Power To Design The Desired

CERTIFICATE

OF PARTICIPATION

This Certificate is Presented to

Mr./Ms. *Surya Raj*

University Roll No. *191500832*

for successfully completing e-Internship on

SMART HOME AUTOMATION USING IOT

from 1st July 2021 to 31st July 2021

[illegible]

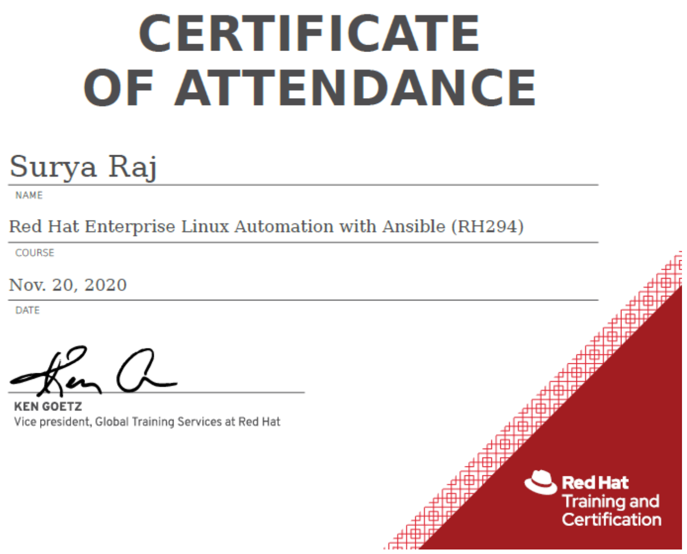
GLA

Authorised Signatory

[illegible]

DCS

Authorised Signatory



CERTIFICATE

OF ATTENDANCE

Surya Raj

NAME

Red Hat Enterprise Linux Automation with Ansible (RH294)

COURSE

Nov. 20, 2020

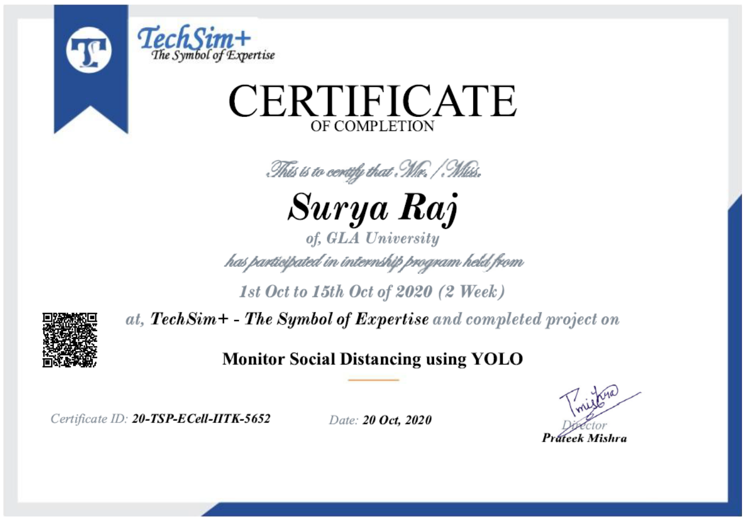
DATE

[illegible]

KEN GOETZ

Vice president, Global Training Services at Red Hat

Red Hat Training and Certification



TechSim+
The Symbol of Expertise

CERTIFICATE
OF COMPLETION

This is to certify that Mr. / Miss.

Surya Raj

of, GLA University

has participated in internship program held from

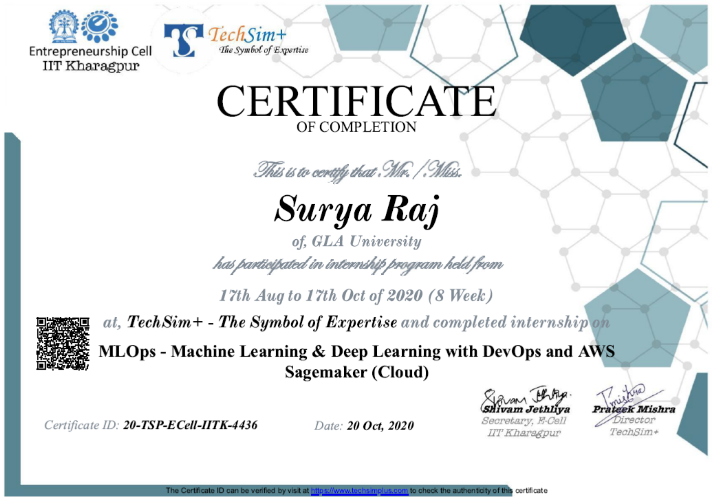
1st Oct to 15th Oct of 2020 (2 Week)

at, ***TechSim+ - The Symbol of Expertise*** *and completed project on*

Monitor Social Distancing using YOLO

Certificate ID: ***20-TSP-ECell-IITK-5652*** *Date:* ***20 Oct, 2020***

Director
Prateek Mishra



Entrepreneurship Cell
IIT Kharagpur

TechSim+
The Symbol of Expertise

CERTIFICATE
OF COMPLETION

This is to certify that Mr. / Miss.

Surya Raj

of, GLA University

has participated in internship program held from

17th Aug to 17th Oct of 2020 (8 Week)

at, ***TechSim+ - The Symbol of Expertise*** *and completed internship on*

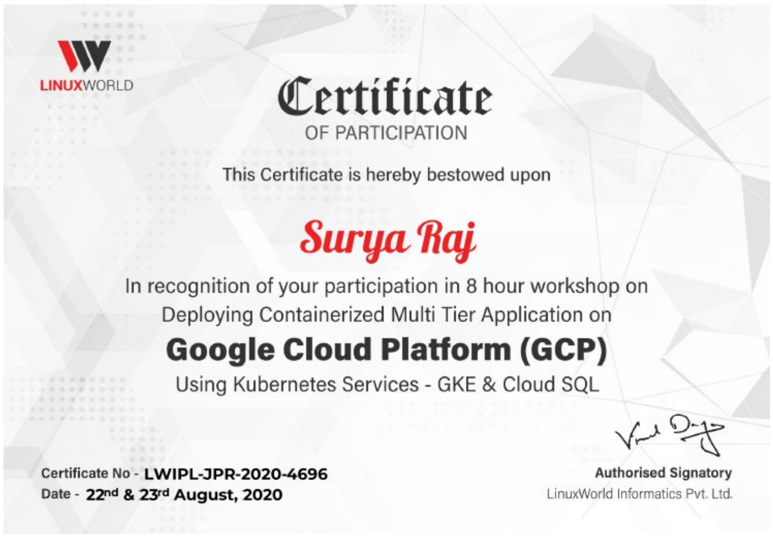
MLOps - Machine Learning & Deep Learning with DevOps and AWS Sagemaker (Cloud)

Certificate ID: ***20-TSP-ECell-IITK-4436*** *Date:* ***20 Oct, 2020***

Shivam Jethliya
Secretary, E-Cell
IIT Kharagpur

Prateek Mishra
Director
TechSim+

The Certificate ID can be verified by visit at https://www.techsimplus.com to check the authenticity of this certificate



LINUXWORLD

Certificate
OF PARTICIPATION

This Certificate is hereby bestowed upon

Surya Raj

In recognition of your participation in 8 hour workshop on
Deploying Containerized Multi Tier Application on
Google Cloud Platform (GCP)
Using Kubernetes Services - GKE & Cloud SQL

Certificate No - **LWIPL-JPR-2020-4696**
Date - **22nd & 23rd August, 2020**

Authorised Signatory
LinuxWorld Informatics Pvt. Ltd.

Department of Computer Engineering and Applications

GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,

Chaumuha, Mathura – 281406 U.P (India)

ACKNOWLEDGEMENT

Presenting the ascribed project paper report in this very simple and official form, we would like to place my deep gratitude to GLA University for providing us the instructor Mr. Mandeep Singh, our technical trainer and supervisor.

He has been helping us since Day 1 in this project. He provided us with the roadmap, the basic guidelines explaining on how to work on the project. He has been conducting regular meeting to check the progress of the project and providing us with the resources related to the project. Without his help, we wouldn't have been able to complete this project.

And at last but not the least we would like to thank our dear parents for helping us to grab this opportunity to get trained and also my colleagues who helped me find resources during the training.

Thanking You

Sign: *Surya Raj*	**Sign**: *Priya Tomar*
Name of Candidate: Surya Raj	**Name of Candidate**: Priya Tomar
University Roll No.:191500832	**University Roll No.**:191500603

ABSTRACT

The basic approach to this project is to make a web application good enough of detecting objects in a single digital image, let it be any particular materialistic thing in the real world, such as helmets, balls, bicycles, etc.

This project would work on Python, Django and some basic Front-end Web Technologies like HTML, CSS with Django framework. To make this project usable, some particular ML techniques will be used for training the datasets and for processing the results. Inside the WebApp, one can submit an image and depending on the ML technique or the kind of detection selected, a backend processing will happen and image would either be processed on a server like AWS cloud or the dedicated system server. The motive behind any detection would be to classify the images, to detect object or whatever we can possibly detect as we train the dataset.

For instance, we can perform object detection where we can detect the instances of semantic objects of certain class (such as humans, buildings, cars, etc.) in digital images.

We can perform facial analysis where we can detect a person's facial expression and display the current emotions of the person, whether the person is smiling, sad, happy or shocked plus their age and gender. With this we can also perform celebrity detection where we can recognize thousands of faces based on ML models and detect the faces in any image like that.

CONTENTS

Cover Page i

Declaration ii

Certificate iii

Training Certificate… iv

Acknowledgement vii

Abstract viii

Content ix

Chapter 1 Introduction 1

- 1.1 Context… 1
- 1.2 Motivation.. 1
- 1.3 Objective.. 2
- 1.4 Existing System 3
- 1.4 Sources.. 5

Chapter 2 Software Requirement Analysis... 6

- 2.1 Requirement Analysis.. 6
- 2.2 Problem Statement… 7
- 2.3 Hardware and Software Requirements 8

Chapter 3 Software Design… 9

- 3.1 Use Case Diagram 10
- 3.2 Data Flow Diagram 11

Chapter 4 Technology Used…..12

- 4.1 DJANGO… 12
- 4.2 REST API 14
- 4.3 AWS REKOGNITION... 16
- 4.4 Tools and Technologies………………………………………………...17

Chapter 5 Implementation.. 18

- 5.1 Implementation of Object Detection.. 18

Chapter 7 Conclusion...20

CHAPTER-1

INTRODUCTION

1.1 CONTEXT

This project “Object Detection” has been submitted in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering at GLA University, Mathura, supervised by Mr. Mandeep Singh. This project has been completed in approximately three months and has been executed in modules, for which meetings have been organized to check the progress of the work and for instructions and guidelines.

1.2 MOTIVATION

The main reason behind such project was to make full use of our already existing technology and bring out a new methodology in which we can integrate some very powerful technologies and bring out a fascinating result. This result would be a webapp that does not need to be installed on any smartphone and can be assessed by the user through any web browser itself. This makes the user skip the process of installing the application , saves memory and provides full functionality as any normal application would.

These days more and more companies or organizations wish for Web Applications which helps them boost their work in many ways. Web applications give businesses the ability to streamline their operations, increase efficiency and reduce costs. These applications provide the same functionality online as their desktop versions would.

Markets like Europe and Asia rely on mobile web apps for security reasons and since users don’t update web apps , immediate improvements can occur to connect bugs and security issues.

1.3 OBJECTIVE

The main objective of this project is to create a web application that will take a digital image as input and detect the objects available in the image and detect the faces as well. With the face you can even detect the identity of the person by displaying their names over their face. Their faces would be highlighted inside a box which would be detecting their faces. So will the object and its type would be mentioned over the object in the image itself.

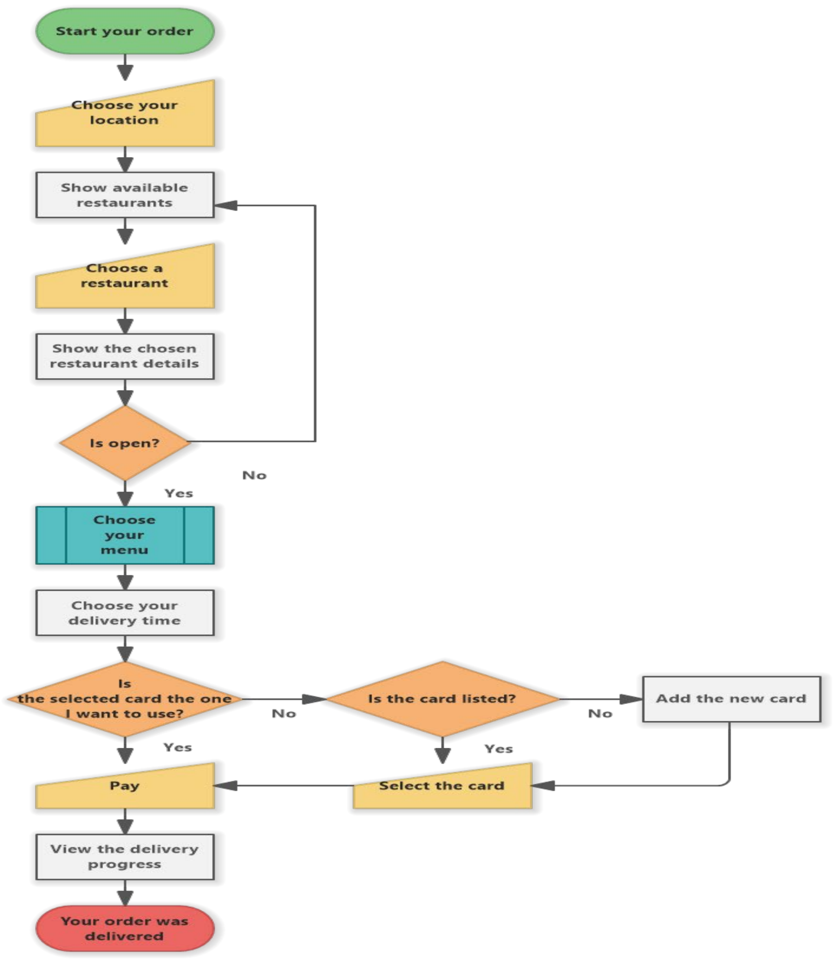
With the faces, one can also tell the current emotions of the person in the given image and determine the age as well. Such methods will all be available inside a browser and no external application has to be downloaded for this use-case.

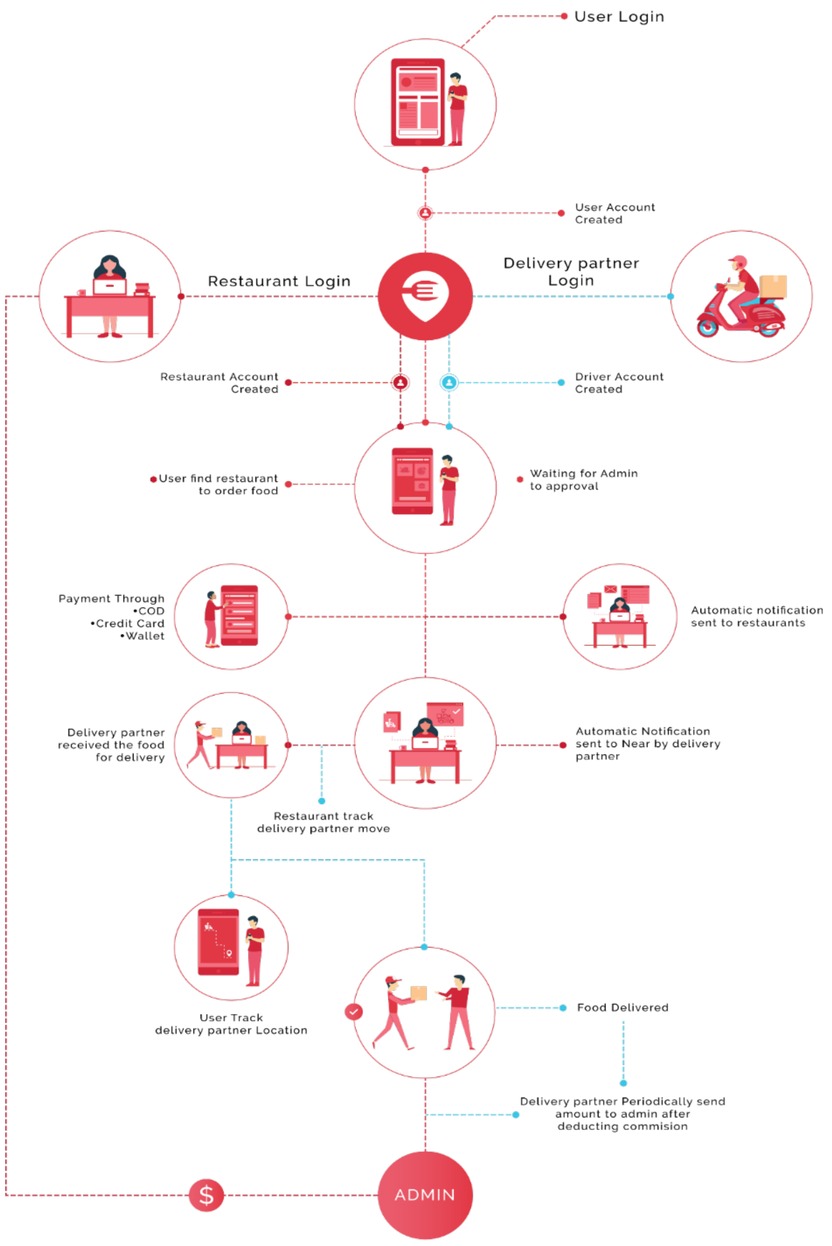
1.4 EXISTING SYSTEM

There already exists systems which are using the Web Technologies and have an API integrated in them. The count of such applications would be immense as so many companies are using this technology model.

For example, Netflix and YouTube use ML to personalize your experience by recommending you new content based on your viewing history. This helps them make better decisions and understand the browsing behaviors of their users.

For instance, most food delivery models like Zomato uses such analogy in these following features :-




User Login
User Account Created
Restaurant Login
Delivery partner Login
Restaurant Account Created
Driver Account Created
User find restaurant to order food
Waiting for Admin to approval
Payment Through •COD •Credit Card •Wallet
Automatic notification sent to restaurants
Delivery partner received the food for delivery
Automatic Notification sent to Near by delivery partner
Restaurant track delivery partner move
User Track delivery partner Location
Food Delivered
Delivery partner Periodically send amount to admin after deducting commision
$
ADMIN

But depending upon the work specified , such as we are using our API to do the Machine Learning work , it may not be specifically available in all Web Applications . The main use of REST API in this project is to deploy Machine Learning model in this project.

AWS Rekognition is another cloud based software as a service platform , which we are using through API service provided by AWS and do the facial analysis.

It is already pre trained and uses deep neural network models to detect and label thousands of objects and scenes in your image.

1.5 SOURCES

The source for flow chart of the given food delivery model https://merehead.com/blog/build-app-like-zomato/

The source of Amazon rekognition

ion https://aws.amazon.com/rekognition/faqs/

CHAPTER -2

SOFTWARE REQUIREMENT ANALYSIS

2.1 Requirement Analysis

The need for an application that can run on any device is unavoidable today due to many reasons such as incompatibility of device software, lack of memory, unnecessarily occupied storage, etc.

Web applications are essential in such scenarios for several reasons. They offer many advantages , the most important being they perform all necessary functions utilizing a web browser instead of a installed software. At the same time , they have cloud based functionality becoming an essential component of business in today's expanding world. Organizations are creating and embracing web applications to meet their business demands.

Part of their importance comes from businesses achieving a competitive edge using web applications. Adding to the power of mobile, web apps are an essential tool for garnering and keeping customers. Since web apps are Internet-enabled apps accessible through the mobile device's web browser, this savvy method gives businesses powerful resources to use on smartphones to market goods and services.

Web applications eliminate the concerns of whether or not the app works effectively and efficiently on numerous platforms, such as desktop, tablet, and other mobile devices. Its cross-platform capability makes web apps no longer a luxury, but in some cases, a necessity. They can be monetized with membership fees, dues, or advertisements, plus, update themselves without the need for user intervention.

Web apps avoid app store memberships, fees, and restrictions when it comes to each software program. They allow businesses to release their versions, on their own time, and steer away from app stores. Additionally, there are all kinds of cloud-based tools allowing businesses to use the cloud for expanding storage.

All businesses need to be aware of security threats with their applications. Web apps can protect websites and software programs. They are designed for more privacy and a high level of security. Markets like Europe and Asia rely on mobile web apps for security reasons and since users don't update web apps, immediate improvements can occur to correct bugs and security issues.

2.2 PROBLEM STATEMENT

The need for a system that can perform tasks on a browser inside mobile applications is immense as let's say one user wants to watch Netflix and he has a smartphone with Android version 4-5, RAM – 2GB and Hard Disk storage available is just 100-200 MB. Normally then , a person will have to install a mobile application in his Android device from Play Store or particular source and it will require the following :-

1. Memory space to install Netflix Application
2. Storage space to store binaries, cache, app data , etc.
3. Extra installation time
4. Extra Network/Internet data to download/install the app.

And normally with such specifications in the Android device, one would not be able to install the application in the first place because it would never meet the requirements to install the application.

But let's say the user has a web browser inside the device and then he can just open the Netflix website, do the same process as he would inside an application and login with his credentials, go to his favorite media file and stream it on his phone itself with the help of Web Browser only. So he did not has to go through the hassle of installing a specific app to meet the criteria.

But the above example was just for reference to show how the requirements often make the needs turn obsolete when they aren't met.
For this project, to find certain objects or features inside an image , we need a compact and easy to use Webapp which can work inside a Web Browser itself.

This will provide many functionalities or same as would a local application.

2.3 HARDWARE AND SOFTWARE REQUIREMENTS

Hardware Requirements to run this application

2.3.1 Processor : intel i3(minimum)
2.3.2 Operating System : Any Operating System with a browser
2.3.3 RAM : 1 GB (minimum)
2.3.4 Hard Disk : 100MB Free space

Software Requirements to create this project

2.3.5 Software used: AWS CLI, Python 3.10, Django, Django_REST_Framework, PyCharm and their dependency libraries
2.3.6 Language used : Python, HTML , JavaScript
2.3.7 API : Django_REST API

CHAPTER- 3
SOFTWARE DESIGN

3.1 USE-CASE DIAGRAM:

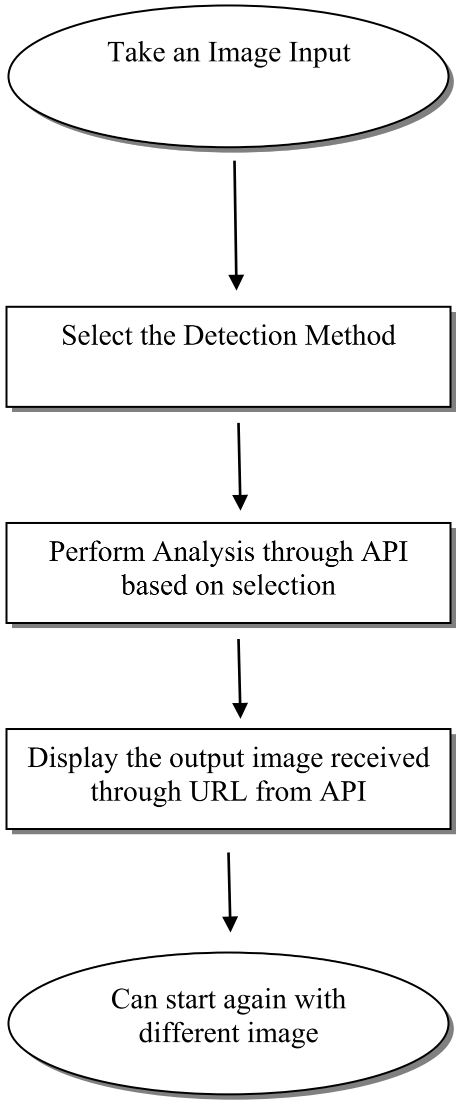

Take an Image Input
Select the Detection Method
Perform Analysis through API based on selection
Display the output image received through URL from API
Can start again with different image

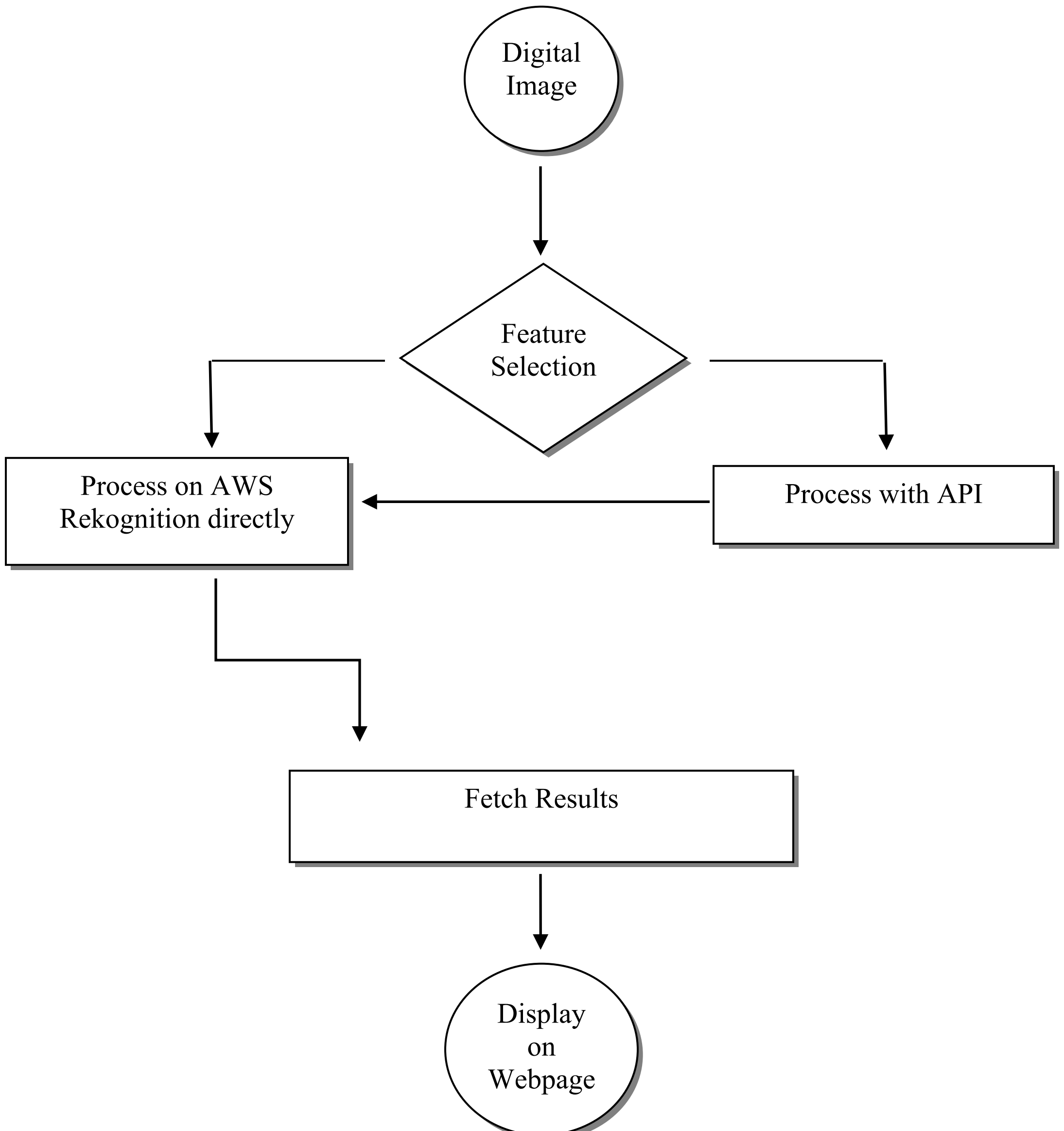
Figure-2: Use–Case Diagram

So the above diagram shows how we are implementing the object detection inside our whole project. The diagram is pretty simple to understand as it shows the steps related to whole process.

On a more deeper end , the data is traveling to a lot more places than just staying in the user's system. That will be shown in the data flow diagram.

3.2 DATA FLOW DIAGRAM

The system works pretty well organized.



CHAPTER-4
TECHNOLOGY USED

4.1 DJANGO

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support.

Django helps you write software that is:

- **Complete**

  Django follows the "Batteries included" philosophy and provides almost everything developers might want to do "out of the box". Because everything you need is part of the one "product", it all works seamlessly together, follows consistent design principles, and has extensive and up-to-date documentation.

- **Versatile**

  Django can be (and has been) used to build almost any type of website — from content management systems and wikis, through to social networks and news sites. It can work with any client-side framework, and can deliver content in almost any format (including HTML, RSS feeds, JSON, XML, etc). The site you are currently reading is built with Django!

Internally, while it provides choices for almost any functionality you might want (e.g. several popular databases, templating engines, etc.), it can also be extended to use other components if needed.

- **Secure**

  Django helps developers avoid many common security mistakes by providing a framework that has been engineered to "do the right things" to protect the website automatically. For example, Django provides a secure way to manage user accounts and passwords, avoiding common mistakes like putting session information in cookies where it is vulnerable (instead cookies just contain a key, and the actual data is stored in the database) or directly storing passwords rather than a password hash.

  A password hash is a fixed-length value created by sending the password through a cryptographic hash function. Django can check if an entered password is correct by running it through the hash function and comparing the output to the stored hash value. However due to the "one-way" nature of the function, even if a stored hash value is compromised it is hard for an attacker to work out the original password.

  Django enables protection against many vulnerabilities by default, including SQL injection, cross-site scripting, cross-site request forgery and clickjacking (see Website security for more details of such attacks).

- **Scalable**

  Django uses a component-based “shared-nothing” architecture (each part of the architecture is independent of the others, and can hence be replaced or changed if needed). Having a clear separation between the different parts means that it can scale for increased traffic by adding hardware at any level: caching servers, database servers, or application servers. Some of the busiest sites have successfully scaled Django to meet their demands (e.g. Instagram and Disqus, to name just two).

- **Maintainable**

  Django code is written using design principles and patterns that encourage the creation of maintainable and reusable code. In particular, it makes use of the Don't Repeat Yourself (DRY) principle so there is no unnecessary duplication, reducing the amount of code. Django also promotes the grouping of related functionality into reusable "applications" and, at a lower level, groups related code into modules (along the lines of the Model View Controller (MVC) pattern).

4.2 REST API

A REST API (also known as RESTful API) is an application programming interface (API or web API) that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services. REST stands for representational state transfer and was created by computer scientist Roy Fielding.

REST is a set of architectural constraints, not a protocol or a standard. API developers can implement REST in a variety of ways.

When a client request is made via a RESTful API, it transfers a representation of the state of the resource to the requester or endpoint. This information, or representation, is delivered in one of several formats via HTTP: JSON (Javascript Object Notation), HTML, XLT, Python, PHP, or plain text. JSON is the most generally popular file format to use because, despite its name, it's language-agnostic, as well as readable by both humans and machines.

Something else to keep in mind: Headers and parameters are also important in the HTTP methods of a RESTful API HTTP request, as they contain important identifier information as to the request's metadata, authorization, uniform resource identifier (URI), caching, cookies, and more. There are request headers and response headers, each with their own HTTP connection information and status codes.

In order for an API to be considered RESTful, it has to conform to these criteria:

A client-server architecture made up of clients, servers, and resources, with requests managed through HTTP.
Stateless client-server communication, meaning no client information is stored between get requests and each request is separate and unconnected.
Cacheable data that streamlines client-server interactions.
A uniform interface between components so that information is transferred in a standard form. This requires that:
resources requested are identifiable and separate from the representations sent to the client.

Resources can be manipulated by the client via the representation they receive because the representation contains enough information to do so.
self-descriptive messages returned to the client have enough information to describe how the client should process it.

hypertext/hypermedia is available, meaning that after accessing a resource the client should be able to use hyperlinks to find all other currently available actions they can take.

A layered system that organizes each type of server (those responsible for security, load-balancing, etc.) involved the retrieval of requested information into hierarchies, invisible to the client.
Code-on-demand (optional): the ability to send executable code from the server to the client when requested, extending client functionality.
Though the REST API has these criteria to conform to, it is still considered easier to use than a prescribed protocol like SOAP (Simple Object Access Protocol), which has specific requirements like XML messaging, and built-in security and transaction compliance that make it slower and heavier.

In contrast, REST is a set of guidelines that can be implemented as needed, making REST APIs faster and more lightweight, with increased scalablity—perfect for Internet of Things (IoT) and mobile app development.

4.3 AWS REKOGNITION

Amazon Rekognition makes it easy to add image and video analysis to your applications using proven, highly scalable, deep learning technology that requires no machine learning expertise to use. With Amazon Rekognition, you can identify objects, people, text, scenes, and activities in images and videos, as well as detect any inappropriate content. Amazon Rekognition also provides highly accurate facial analysis and facial search capabilities that you can use to detect, analyze, and compare faces for a wide variety of user verification, people counting, and public safety use cases.

With Amazon Rekognition Custom Labels, you can identify objects and scenes in images that are specific to your business needs. For example, you can build a model to classify specific machine parts on your assembly line or to detect unhealthy plants. Amazon Rekognition Custom Labels takes care of the model development heavy lifting for you, so no machine learning experience is required. You simply need to supply images of objects or scenes you want to identify, and the service handles the rest.

4.4 AWS CLI

The AWS Command Line Interface (CLI) is a unified tool to manage your AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts.

The AWS CLI v2 offers several new features including improved installers, new configuration options such as AWS Single Sign-On (SSO), and various interactive features.

Aws-shell is a command-line shell program that provides convenience and productivity features to help both new and advanced users of the AWS Command Line Interface. Key features include the following.

- Fuzzy auto-completion for
  - Commands (e.g. ec2, describe-instances, sqs, create-queue)
  - Options (e.g. --instance-ids, --queue-url)
  - Resource identifiers (e.g. Amazon EC2 instance IDs, Amazon SQS queue URLs, Amazon SNS topic names)
- Dynamic in-line documentation

  - Documentation for commands and options are displayed as you type
- Execution of OS shell commands
  - Use common OS commands such as cat, ls, and cp and pipe inputs and outputs without leaving the shell
- Export executed commands to a text editor

The AWS Command Line Interface guide walks you through installing and configuring the tool. After that, you can begin making calls to your AWS services from the command line.

```
$ aws ec2 describe-instances






$ aws ec2 start-instances --instance-ids i-1348636c






$ aws sns publish --topic-arn arn:aws:sns:us-east-1:546419318123:OperationsError -
-message "Script Failure"






$ aws sqs receive-message --queue-url
https://queue.amazonaws.com/546419318123/Test
```

You can get help on the command line to see the supported services,

```
$ aws help
```

the operations for a service,

```
$ aws autoscaling help
```

and the parameters for a service operation.

```
$ aws autoscaling create-auto-scaling-group help
```

4.5 TOOLS AND LANGUAGES

Tools and Languages used to build this project are:-

- **PyCharm :** PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development.

- **Python :** Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

- **HTML :** The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript.

- **JavaScript** : JavaScript is a scripting or programming language that allows you to implement complex features on web pages — every time a web page does more than just sit there and display static information for you to look at — displaying timely content updates, interactive maps, animated 2D/3D graphics, scrolling video jukeboxes, etc. — you can bet that JavaScript is probably involved. It is the third layer of the layer cake of standard web technologies after HTML and CSS.

CHAPTER -5

IMPLEMENTATION

Creating a web app concept design with screen sketches and functional flow diagrams is the best way to communicate your vision to the developer. Making the concept clear to the developer is probably the most important factor in successful webapp development. Yet it is one of the most common problems or obstacles in a webapp development outsourcing project.

No matter what the marketing and profit goals are or if you are outsourcing an app for your personal use, you need to fully design and document the app concept if you expect a programmer to make your vision a reality. Developers are not mind readers and even descriptions given during conversations can be very fleeting or interpreted differently. Fully documenting your concept, therefore, leaves little to chance. The two most important things to do are: A) make a comprehensive description of how the app works and what it does (functionality) and B) create a comprehensive description of what the user sees and does (look and feel).

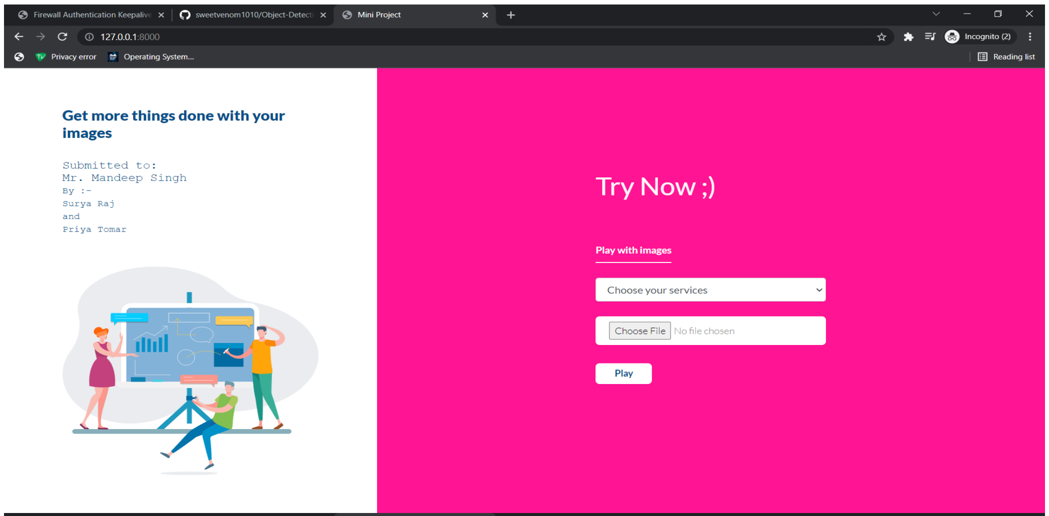
5.1 Implementation of the Object Detection:

Object detection, one of the most fundamental and challenging problems in computer vision which seeks to locate object instances from a large number of predefined categories in natural images. Deep learning techniques have emerged as a powerful strategy for learning feature representations directly from data and have led to remarkable breakthroughs in the field of generic object detection.

The goal of object detection is to determine whether there are any instances of objects from given categories (such as humans, cars, bicycles, dogs or cats) in an image and, if present, to return the spatial location and extent of each object instance (e.g., via a bounding box). As the cornerstone of image understanding and computer vision, object detection forms the basis for solving complex or high level vision tasks such as segmentation, scene understanding, object tracking, image captioning, event detection, and activity recognition. Object detection supports a wide range of applications, including robot vision, consumer electronics, security, autonomous driving, human computer interaction, content based image retrieval, intelligent video surveillance, and augmented reality.

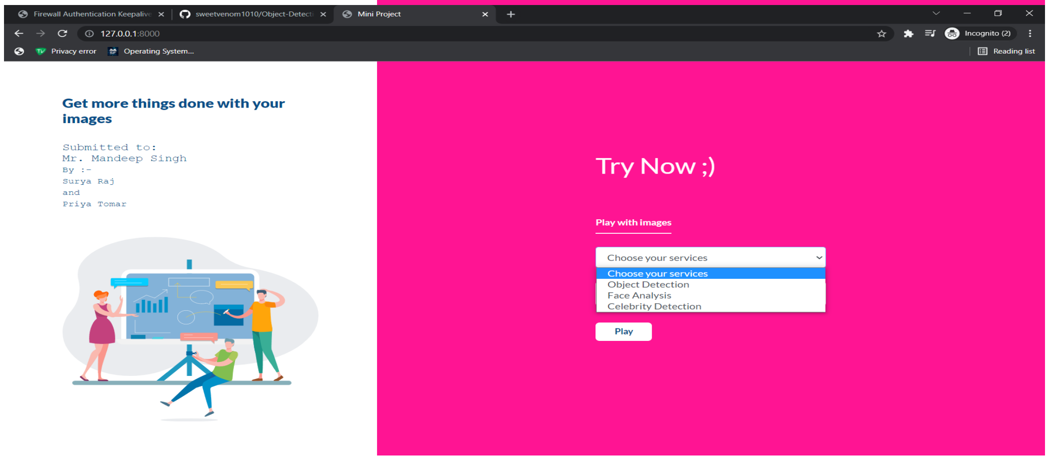
5.2 Snapshots of Implementation :

Snapshot of Web Page / WebApp :



Get more things done with your images

Submitted to:
Mr. Mandeep Singh
By :-
Surya Raj
and
Priya Tomar

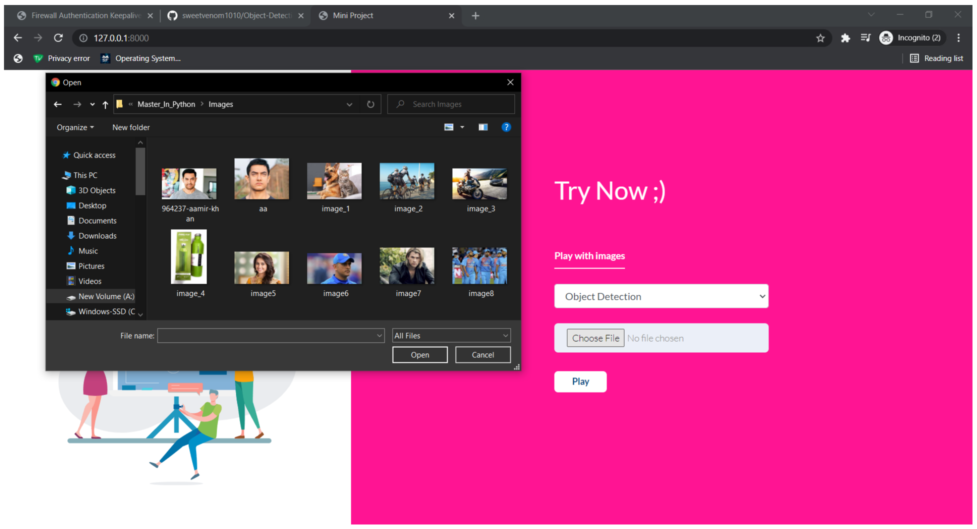


Get more things done with your images

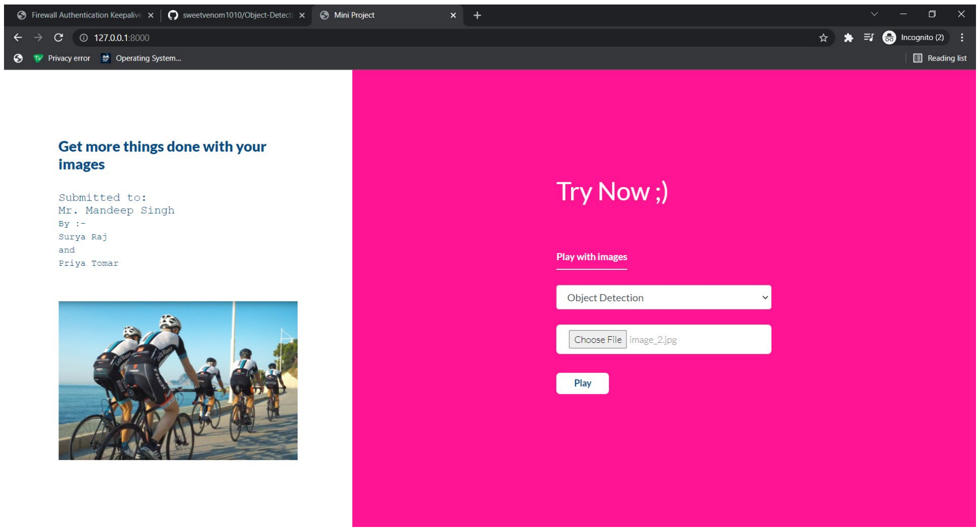
Submitted to:
Mr. Mandeep Singh
By :-
Surya Raj
and
Priya Tomar

Object Detection

Selecting image for the service of Object Detection:



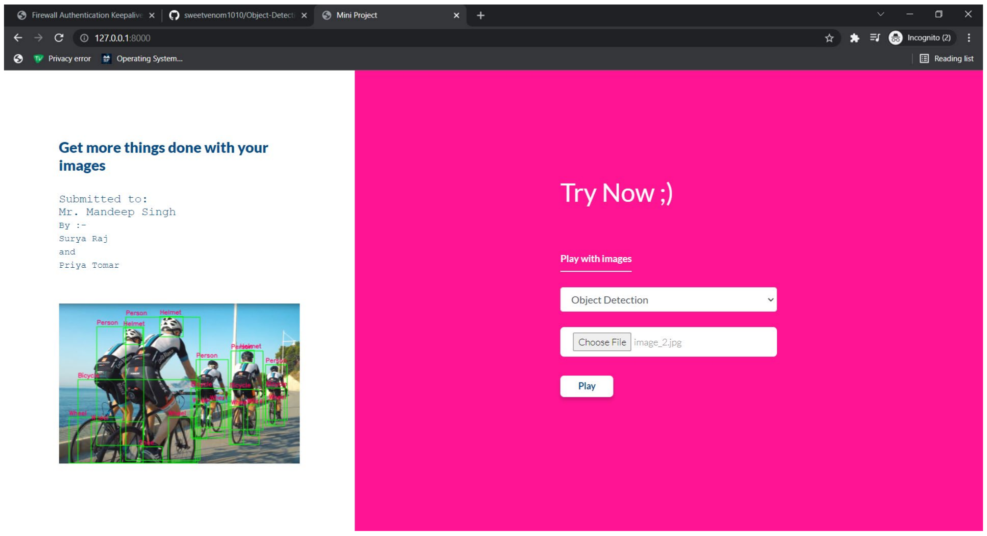
The image gets refreshed to selected image as soon as selected:



Get more things done with your images

Submitted to:
Mr. Mandeep Singh
By :-
Surya Raj
and
Priya Tomar

The objects get updated instantly as soon as API fetches resulted image to Web Page:

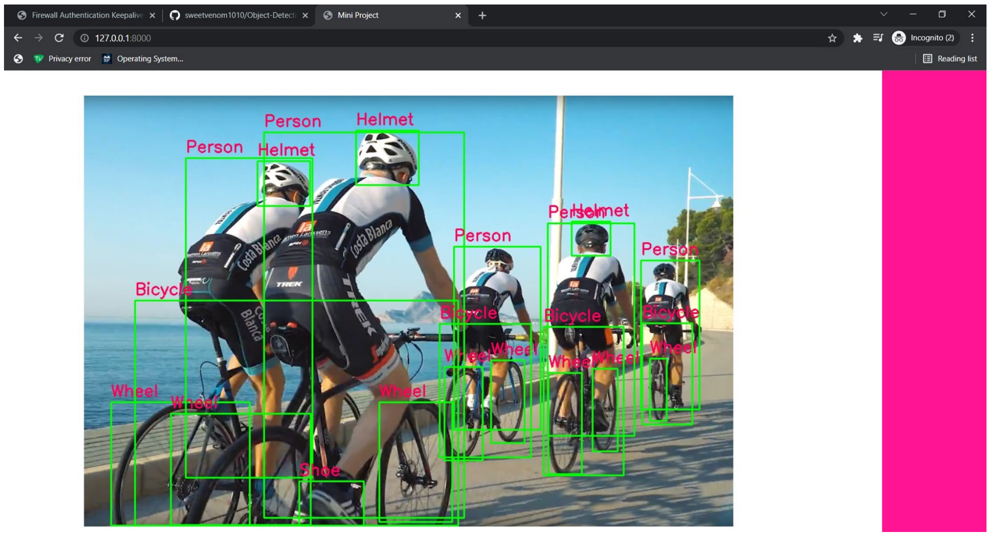


Get more things done with your images

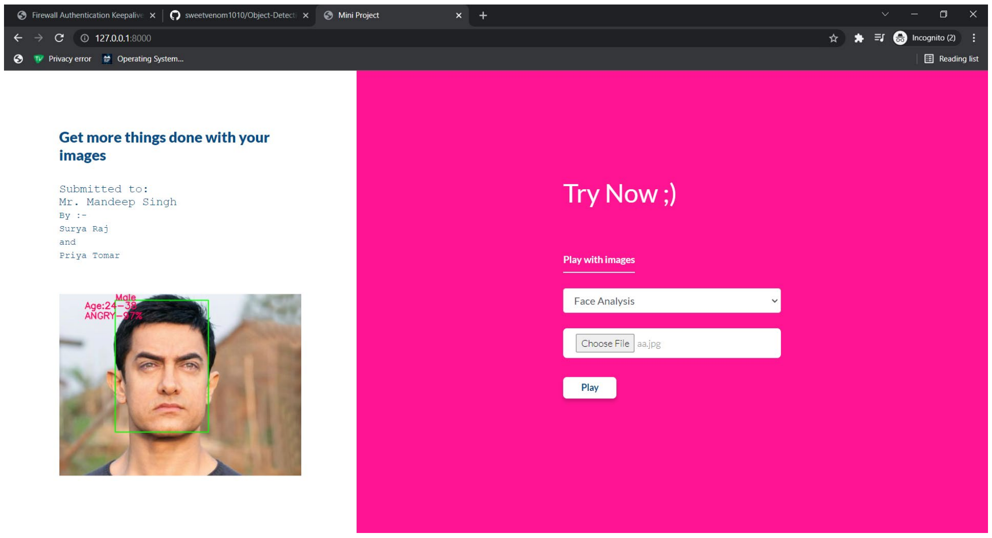
Submitted to:
Mr. Mandeep Singh
By :-
Surya Raj
and
Priya Tomar

Object Detection

(Zoomed Webpage) Bounding Box with detected objects appear as:



Snapshot of Face Analysis Service :



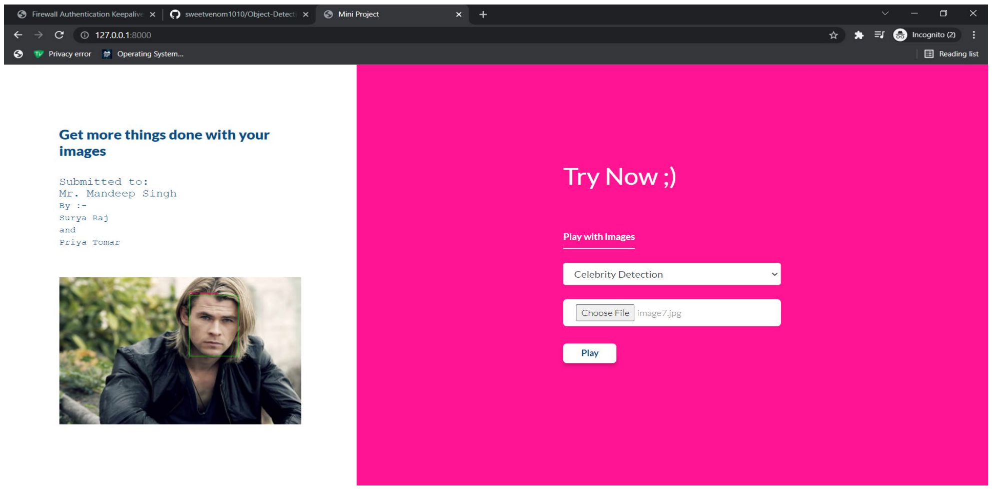
Get more things done with your images

```
Submitted to:
Mr. Mandeep Singh
By :-
Surya Raj
and
Priya Tomar
```

Face Analysis

aa.jpg

Snapshot of Celebrity Detection Service:



Get more things done with your images

```
Submitted to:
Mr. Mandeep Singh
By :-
Surya Raj
and
Priya Tomar
```

Celebrity Detection

image7.jpg

CHAPTER -6
CONCLUSION

Proposed Object Detection is a part of Computer Vision which is necessary for so many applications in our life that it can change the whole prospects about how we live. This project brings a machine close to imparting human vision, understand and look at the environment as a human would and make decisions accordingly by selecting only the required elements from surroundings.

These days Smart camera applications provide a scalable method to implement automated visual inspection and quality control of production processes and assembly lines in smart factories.

Another application field of vision systems is optimizing assembly line operations in industrial production and human-robot interaction. The evaluation of human action can help construct standardized action models related to different operation steps and evaluate the performance of trained workers.

Machine learning is incorporated in medical industries for purposes such as breast and skin cancer detection. For instance, image recognition allows scientists to detect slight differences between cancerous and non-cancerous images and diagnose data from magnetic resonance imaging (MRI) scans and inputted photos as malignant or benign.

Object detection is a key ability for most computer and robot vision system. Although great progress has been observed in the last years, and some existing techniques are now part of many consumer electronics (e.g., face detection for auto-focus in smartphones) or have been integrated in assistant driving technologies, we are still far from achieving human-level performance, in particular in terms of open-world learning.

It should be noted that object detection has not been used much in many areas where it could be of great help. As mobile robots, and in general autonomous machines, are starting to be more widely deployed (e.g., quad-copters, drones and soon service robots), the need of object detection systems is gaining more importance. Finally, we need to consider that we will need object detection systems for nano-robots or for robots that will explore areas that have not been seen by humans, such as depth parts of the sea or other planets, and the detection systems will have to learn to new object classes as they are encountered. In such cases, a real-time open-world learning ability will be critical.