ROBUST SECURITY CAMERA

By

Name	CMS ID
SAHAR BATOOL	41598
HUSSNA ALI	44072
AFTAB HUSSAIN	41699
HASNAIN RAZA	41018

2021-FICT-BS-IT-Fall-17

Supervisor

SIR SIKANDER KHAN

Department of Information Technology

A thesis submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Information Technology (BS-IT)

In

Department of Information Technology, Faculty of Information,

Communication Technology, BUITEMS,

Quetta, Pakistan.

(Sept,2021)

DEPARTMENT OF INFORMATION TECHNOLOGY

ROBUST SECURITY CAMERA

FYP THESIS

SUPERVISOR: SIR SIKANDER KHAN

GROUP MEMBERS

NAME	CMS ID
SAHAR BATOOL	41598
HUSSNA ALI	44072
AFTAB HUSSAIN	41699
HASNAIN RAZA	41018

Undertaking

We certify that research work entitled **“Robust Security Camera”** is our own work. The work has not been presented elsewhere for assessment. Where material has been used from other sources it has been properly acknowledged / referred. This work/idea is property of Baluchistan University of Information Technology Engineering and Management Sciences (BUITEMS), Quetta. All rights reserved and no part of this work/publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of BUITEMS.

Sahar Batool (41598)

Sept,2021

____

Hussna Ali (44072)

Sept, 2021

____

Aftab Hussain (41699)

Sept, 2021

____

Hasnain Raza (41018)

Sept, 202

____

Approval

It is certified that the contents of the thesis entitled **“Robust Security Camera**"

is submitted by

By

Name	CMS ID
SAHAR BATOOL	41598
HUSSNA ALI	44072
AFTAB HUSSAIN	41699
HASNAIN RAZA	41018

have been found satisfactory for the requirement of the degree.

Supervisor: Sir Sikander Khan

Signature: ________________

Date: ________________

FYP Coordinator: Sir Sikander Khan

Signature: ________________

Date: ________________

Abstract

CCTV cameras are almost installed in every private and public sector such as educational institutes malls, hotels, firms, cinema etc. in order to provide security measures. Monitoring multiple screens continuously is quite risky for the single CCTV operator for prolonged time. Another problem is the current situation that peoples around the world are facing, that is COVID-19. Covid-19 pandemic has increased the trend of wearing face mask to secure life as this virus spread from person to person easily. The aim of our project is to train CCTV camera for detection of the weapon (the most common way to harm/ rob anyone) in the image, counting people along-with proper mask detection (most important regarding current situation i.e., Covid-19, as calculated number of people with proper mask application will enter the mall/firm/hotel etc. helping in proper application of SOPs). We used YOLOv3 ML model (with 75% accuracy) for weapon detection and CNN’s Sequential ML model (with 99% accuracy) for Face Mask Detection. We developed two applications i.e. Mobile Application (through android studio) and Web Application (through Visual Studio). We integrated our web application with ML models (Weapon Detection Model & Face Mask Detection Model) using “Flask” as a framework while “Python” as a programming language. And integrated mobile application with web application using “Firebase Cloud Platform” as a framework while “Java” as a programming language.

Dedication

We dedicate our work to Allah Almighty, Thank You for the power of mind, protection, strength, guidance and giving us a healthy life.

Our parents who motivated, inspired, and gave us strength when we thought of giving up, who continually provide their spiritual, emotional, moral, and financial support.

Our supervisor who shared his words of advice and encouraged to finish our work by helping us.

Acknowledgment

We would like to thank Our Supervisor “Sir Sikander Khan” for his valuable guidance and extension of every possible help, and our family for being behind us in difficult phases.

LIST OF CONTENTS

LIST OF FIGURES

CHAPTER:1 INTRODUCTION

1.1 PROBLEM STATEMENT

1.1.1 SECURE LIFE

Life is what matters a lot for each individual. People in every consequence want to live a secure life so, they always prefer using all those software, devices, autonomous systems, or others that provide security among which CCTV (Closed Circuit Television) camera plays an important role. Number of incidents happens all around the world such as robbery, terrorist attack, etc. although CCTV cameras are almost installed in every private and public sectors such as educational institutes malls, hotels, firms, cinema etc. in order to provide security and measures such as alarms, site design and systems etc. which are not enough for fulfilling today’s need as the increase in number of surveillance camera footages is challenge for CCTV operator to monitor, analyze and make a decision about whether a dangerous situation is about to happen or not. Monitoring multiple screens continuously is quite risky for the single CCTV operator for prolonged time.

1.1.2 SPREAD OF COVID-19

Another problem is the current situation that people all around the word are facing, that is COVID-19. The main reason behind the spread of COVID-19 is that SOPs are not followed by most of the people. Covid-19 pandemic has increased the trend of wearing face mask to secure life as this virus spread from person to person easily. In 2020, more than 5 million people are infected by the Covid-19.

1.2 AIM AND OBJECTIVES OF THE PROJECT

The aim of our project is to train CCTV camera for detection of the weapon (the most common way to harm/ rob anyone) in the image, counting people along-with proper mask detection (most important regarding current situation i.e. Covid-19, as calculated number of people with proper mask application will enter the mall/firm/hotel etc. helping in proper application of SOPs) and alerting the CCTV operator about it through web application while security guard through android application(automated alert message will be send from web app to android app)which were not being provided by primitive/general CCTV cameras but we will assure you regarding proper provision of above mentioned additional features in our project by integrating algorithm of object detection from field of the computer vision and processing of image through automated surveillance.

1.3 PROJECT SCOPE

The purpose of our project is not only security but business satisfaction too as from detection of weapon and proper application of SOP'S (counting people & entering them and face mask detection) till cost effective integration and deployment of CCTV cameras; with which they won't need to change their cameras rather bringing changes through integration.

It will be based on reviewing and working on ML Models i.e., weapon detection, face mask detection and counting people) and will be consisting of two applications i.e., web-application (highlighting the screen for CCTV operator along-with generating an automated alert message) and android-application (for alerting the guards).

Selecting such a project will save cost regarding time, money and labor work as all you need is to integrate all the features in CCTV camera and not to deploy each and every individual system. So, will not only satisfy customer regarding security but will shape business satisfaction too.

1.4 BENEFITS/UTILITY OF THE PROJECT

CCTV cameras are helpful for all of us so, why not to improve them even more? As we know a lot of CCTV cameras are installed in both private and public sectors and videos of these CCTVs are monitored by a single operator. Thus, it is quite difficult and risky for that person to monitor multiple screens for prolonged time. But through artificial intelligence we can improve the features of CCTV cameras and make it easy for CCTV operator(s) for suspecting the screen with suspicious activity.

Combination of the automatic detection of the objects with risk-based screening can be particularly beneficial. The Weapon Detection Algorithm helps detect weapons such as gun, knife etc. at real time and automatically sends alarm message(s) to control room that makes it easier for screen operator to focus on the screen in which any suspicious activity is taking place and he/she gets opportunity to react quickly.

Covid-19 has become the headline of news since 2019 and people from every platform wants to contribute from scientists to tech specialists. Based upon the measures to be taken for its prevention, face mask is the important one as per sources it reduces the spread of virus thus disease.

Counting people is one of most important features regarding the current situation i.e., Covid-19 that spreads between the people through direct surfaces, objects or via close contact (even within 1m) with the infected person through nose or mouth secretions. Thus, It is really important to allow limited people in any area such as malls, hotels and firms etc. to keep everyone safe because health is what matters a lot. If limited people are allowed in malls, firms or hotels with proper implementations of SOPs then there are less chances for Covid-19 to spread.

1.5 COMMERCIALIZATION/BUSINESS POTENTIAL OF THE PROJECT

Our project has the potential of both businesses to consumer and business to business provisions as through this we can not only serve the community by providing security but will help businesses to expand and get extended more and more because you won’t need to buy or install any new CCTV camera rather integrating our discussed features in a single CCTV camera and it would also be customized as per business need(s) i.e. whether they want to integrate a single feature or all.

As life matters a lot both public and private sectors are ready to invest in such a project. Like; educational institutes, malls, hotels, firms, aviation industries, banks and commercial spaces as these are not only threatened by terrorists, anti-social groups but humans (person to person virus transmission) too.

It will be launched in Pakistan but has the potential to get extended in other countries too as all and all everyone wants to live a secure life.

CHAPTER:2 LITERATURE REVIEW

2.1 WEAPON DETECTION

A system for detecting concealed weapons in an X-ray image has been proposed by Roomi and Rajashankarii. This system uses the fuzzy KNN (KNearest Neighbors) to classify objects as a threat or non-threat. Multiple objects are extracted by using shape-based image segmentation method and feature extraction methods. The image is converted to binary image by choosing the threshold as the mean of the two peaks of bimodal histogram and the objects are labelled. The area of each object is computed, and the values are sorted. The mean value is calculated and is set as threshold to collect the objects of interest. Object boundary is extracted, and shape feature extraction algorithm is implemented. The classifier is trained with the extracted features and the object is classified as object or non-object[1].

A real-time situational recognition and awareness from CCTV image analysis has been proposed. This system supports the CCTV by automatically detecting a “Dangerous Object” and raising an alarm. Detection is performed by a pre trained neural network. The proposed algorithm utilizes an MPEG-7 classifier in cascade with the artificial neural network in order to decrease the number of false positives, this algorithm has a very low detection sensitivity while giving a 100% specificity[2].

A method using CNN Tensor flow-based implementation for detecting and classifying weapons in images has been proposed by Lai and Maples. The researchers used over 1.3 million images with approximately 3000 weapon-based images. A large amount of training data is selected so that to include every situation and orientation a weapon might show up on CCTV footage[3].

In the year 2018, Olmos et al proposed an automatic system for detecting handguns from CCTV videos. The researchers attempted to reduce the number of false detections by using a Deep Convolutional Neural Network (CNN) classifier. The objective of this research is to implement an automated and real-time firearm detection algorithm that assists a human operator by raising an alarm whenever a potentially threatening object (gun) is detected. To get the desired outcome the algorithm must be fully automatic, operate in real-time, accurate, and run on a standard personal computer[4].

A system of surveillance for suppressing mass shootings (as 100000 killed in Pakistan every year), robbery, & break-in using YOLOv3 was proposed by Sanam, Bishwajeet, Doris, Ciro and Rizwan which not only detect the guns, rifles, firearm & others but was keeping locations of incidents as well, consisting of camera, CCTV operator & security panels[5].

2.2 FACE MASK DETECTION

Covid-19 pandemic has increased the trend of wearing face mask to secure life as this virus spread from person to person easily. In 2020, more than 5 million people are infected by the Covid-19. Thus, in this research paper we work on a model that support both image processing and machine learning techniques.

The system is integration of both machine learning and deep learning techniques with Tensor Flow, Open CV and Keras to identify the person with or without Mask on video/image stream.

For training the system, first step is to train the Mobile NetV2 (Deep Learning Model) and then apply detector of face mask over video/images live stream[6].

Detecting Face Mask is challenging task because of two reasons; lack of datasets of the masked faces and facial cues absence from masked regions. To overcome these issues; this paper present dataset as MAFA with 30,811 images of internet and 35,806 masked faces from the social networks such as flicker and search engine for image such as Bing and Google.

Then for mask detection we propose the LLE-CNNs that consist of the three major modules. The proposed module combines the CNNs first to extract the region of facial candidate from input images and then represent them with descriptors of high dimensional. Then embedding module is integrated for turning on descriptors into the similarity-based descriptors with the help of LLE (locally linear embedding) dictionaries and algorithm that is trained on large pool of the synthesized masked faces, normal faces and non-faces. And at last verification module is integrated for identifying the facial regions of candidate and clarify their position by jointly performing the regression and classification tasks within the unified tasks[7].

The healthcare system is facing many crises due to COVID-19. Many precautionary measures have been taken to reduce spread of COVID-19 where one of them is wearing mask.

To detect that whether the person is wearing face mask or not “face mask detection” technique is used that is similar to the object detection from any scene. Architecture of Deep Learning has shown its role in the object detection and this architecture can be used to detect the face mask.

The dataset consists of 1539 images in total that are divided into two sets as “training” and “test” set. 20% dataset is used for testing and 80% dataset is used for training. We train architecture for 100 epochs as training further cause the overfitting on training data. Overfitting occurs when model learns unwanted patterns of training dataset. Thus, accuracy of training increases and accuracy of test decreases. The model that is trained showed the accuracy of 98.7% and 0.985 AUC on unseen test dataset[8].

Covid-19 has become the headline of news since 2019 and people from every platform wants to contribute from scientists to tech specialists.

Based upon the measures to be taken for its prevention, face mask is the important one as per sources it reduces the spread of virus thus disease. In order to accomplish this task Vinita and Velantina proposed a model that not only detects the faces but masks too.

The pre-labelled images as “mask” and “no mask” had been used in order to feed ML models correctly and two OpenCV models i.e. (Floating point 16 version of the original Cafee implementation and 8-bit quantised version using TensorFlow for face detection) while Pytorch library for MobileNetV2 model implementation for image classification as “mask” or “no mask” had been implemented[9]**.**

2.3 COUNTING PEOPLE

According to Walter and Horst research counting of people can be divided into two techniques by using the neural based estimation of crowd and methods that are based on blob tracking and blob detection.

There are two constraints that can be used for evaluation of the current counting people techniques:

1)The counting people system must yield if no person is in viewed area.

2) There are maximum people limit n_{max} for each viewing area and counting people system shall not exceed that value.

Based on the estimators of neural network constraint one is critical point for the counting people methods. As image processing for the feature extraction do not classify the blobs or edges at all thus, coping with first constraint is hard. That's why these are sensible to the other objects rather people and shadows illumination changes.

Fulfilling both constraints by blob technique depends on the robustness and reliability of the realized blob classification and blob detection. As blob is analyzed based on their shape and size, thus, blob technique has the ability to separate people from the other objects[10].

Based on real life video accomplished by CCTV camera people counter so far is the most important topic to be discussed. Among lots of existing technologies, the approach of Zahra Salah's is the mixture of some out of those. As per her approach, Background Segmentation is used for separating background and foreground realties, Object detection for face detection, Binarization for converting color images to binary for minimizing data's length, while Erosion and Dilation processes to erase the noise (If an image contains)[11].

Yasemin & Selcuk, based on real-time and directional counting proposed an algorithm using the Graphic Processing Unit (GPU) Programming for the purpose of detecting and counting people. The obtained experimental results show that their proposed algorithm running on GPU can be successfully programmed and implemented for people detecting and counting problems[12].

Counting people is important not only for businesses but security purposes too. S. Saxena & D. Songara introduced an automatic people counting system which can count multiple people, by using only one camera. Their algorithm uses Viola Jones method of facial recognition to detect people; using a single overhead mounted camera, it counts the number of people going into an observed area. Counting is performed by analyzing the image to detect faces and the system achieves correct people counting rate of 85%.

CHAPTER:3 METHODOLOGY

3.1 INTRODUCTION

Security is amongst the basic necessities of life, and we all deserve to have it. Thus, our project is based-on security which aims to train CCTV camera for detection of the weapon (the most common way to harm/ rob anyone) in the image, counting people along-with proper mask detection (most important regarding current situation i.e. Covid-19, as calculated number of people with mask will enter the mall/firm/hotel etc. helping in proper application of SOPs) and alerting the CCTV operator about it through web application while security guard through android application(automated alert message will be send from web app to android app) that will be done by integrating algorithm of object detection from field of the computer vision and processing of image through automated surveillance.

3.1.1 WEAPON DETECTION MODEL

For Weapon Detection Model we went through various research papers such as **Roomi's & Rajashankarii's** system for detecting concealed weapons in an X-ray image using fuzzy KNN (KNearest Neighbors) to classify objects as a threat or non-threat, **Mansoor's** real-time situational recognition and awareness from CCTV image analysis using pre-trained neural network that utilizes an MPEG-7 classifier in cascade with the artificial neural network in order to decrease the number of false positives, this algorithm has a very low detection sensitivity while giving a 100% specificity, **Sanam's , Bishwajeet's , Doris's, Ciro's and Rizwan's** system of surveillance for suppressing mass shootings (as 100000 killed in Pakistan every year), robbery, & break-in using YOLOv3 and out of the above mentioned ML models we worked on YOLOv3.

We gathered datasets (Positive & Negative) from different sources i.e., Github & Kaggle, labelled them individually using “label img”, used YOLOv3 model for training and testing and got the accuracy as 75%.

3.1.2 FACE MASK DETECTION MODEL

For Face Mask Model we went through various research papers such as **Bhadani’s & Sinha’s** system which is the integration of both machine learning and deep learning techniques with Tensor Flow, Open CV and Keras to identify the person with or without Mask on video/image stream, **Shiming Ge’s, Jia’s, Qiting’s & Zhao’s** proposed LLE-CNNs that consist of the three major modules for mask detection. The proposed module combines the CNNs first to extract the region of facial candidate from input images and then represent them with descriptors of high dimensional, **Vinitha’s & Velantina’s** used pre-labelled images as “mask” and “no mask” had been used in order to feed ML models correctly and two OpenCV models i.e. (Floating point 16 version of the original Cafee implementation and 8-bit quantised version using TensorFlow for face detection) while Pytorch library for MobileNetV2 model implementation for image classification as “mask” or “no mask” had been implemented**,** out of which we used Sequential CNN model.

We gathered datasets (Mask and Unmask) from different sources i.e., Github & Kaggle, used CNN ‘s Sequential model through which we got labelled images via pre-processing process, trained model with 50 epochs, tested model and got accuracy of 99%.

3.1.3 PEOPLE COUNTER

For People Counter we went through various research papers such as **Walter's and Horst's** research counting of people can be divided into two techniques by using the neural based estimation of crowd and methods that are based on blob tracking and blob detection, **Yasemin's & Selcuk's** algorithm based on real-time and directional counting using the Graphic Processing Unit (GPU) Programming for the purpose of detecting and counting people, **S. Saxena's & D. Songara's** introduced an automatic people counting system which can count multiple people, by using only one camera. Their algorithm uses Viola Jones method of facial recognition to detect people; using a single overhead mounted camera, it counts the number of people going into an observed area.

We used flag in order to incorporate counter's code in Face Mask Detection ML model that counts both people with mask & without mask separately.

3.1.4 INTEGRATION

We integrated our web application with ML models (Weapon Detection Model & Face Mask Detection Model) using "Flask" as a framework while "Python" as a programming language.

And integrated mobile application with web application using "Firebase Cloud Platform" as a framework while "Java" as a programming language.

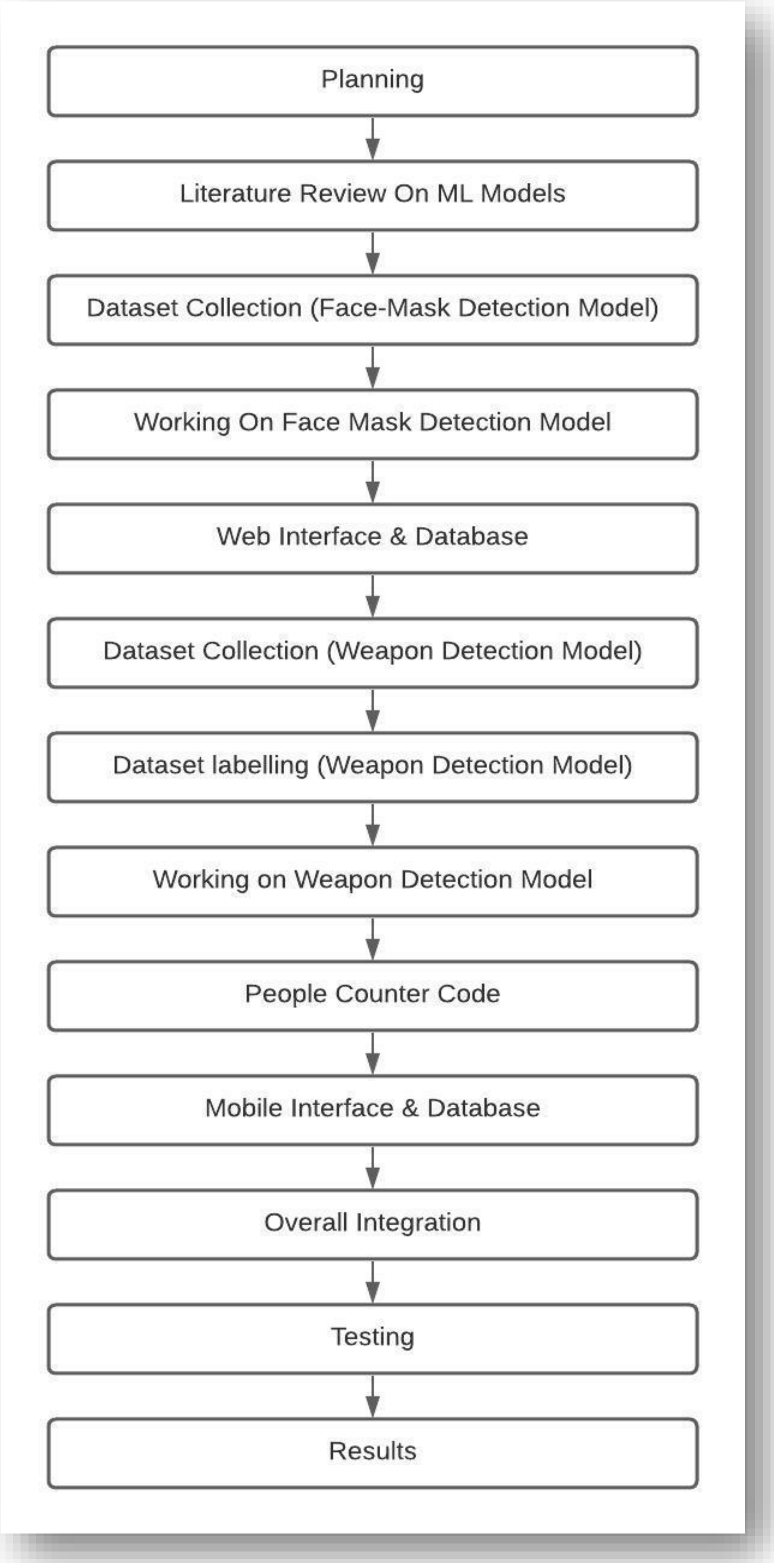

Planning
Literature Review On ML Models
Dataset Collection (Face-Mask Detection Model)
Working On Face Mask Detection Model
Web Interface & Database
Dataset Collection (Weapon Detection Model)
Dataset labelling (Weapon Detection Model)
Working on Weapon Detection Model
People Counter Code
Mobile Interface & Database
Overall Integration
Testing
Results

Figure 1 METHODOLOGY

3.2 PLANNING

A goal without a plan is just a wish so to fulfill the needs of our project we distributed tasks among all of us and started working on it i.e., the Gantt Chart & Work Distribution Chart.

3.2.1 GANTT CHART

- ❖ **PHASE-I**

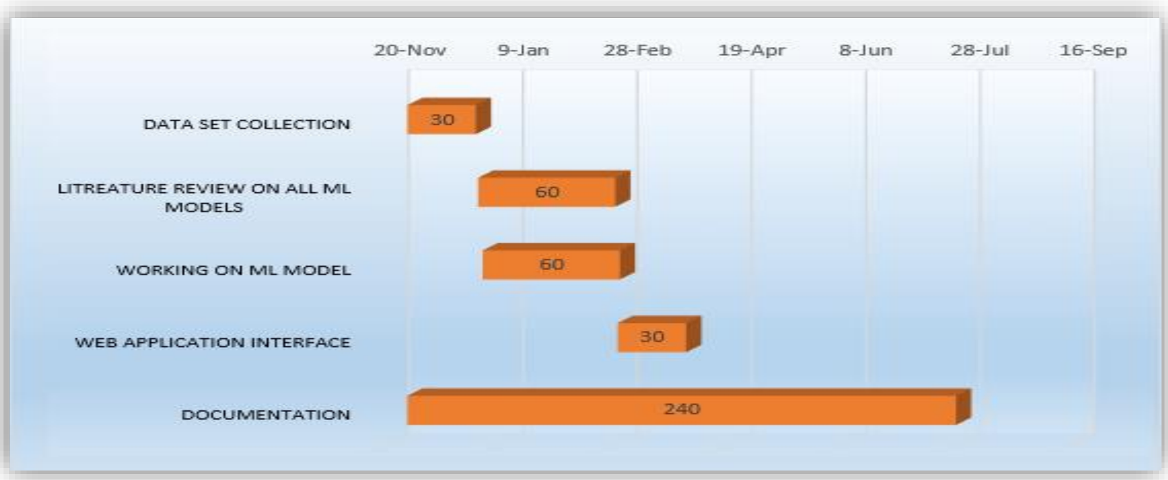


Figure 2 GANTT CHART (PHASE-I)

- ❖ **PHASE-II**

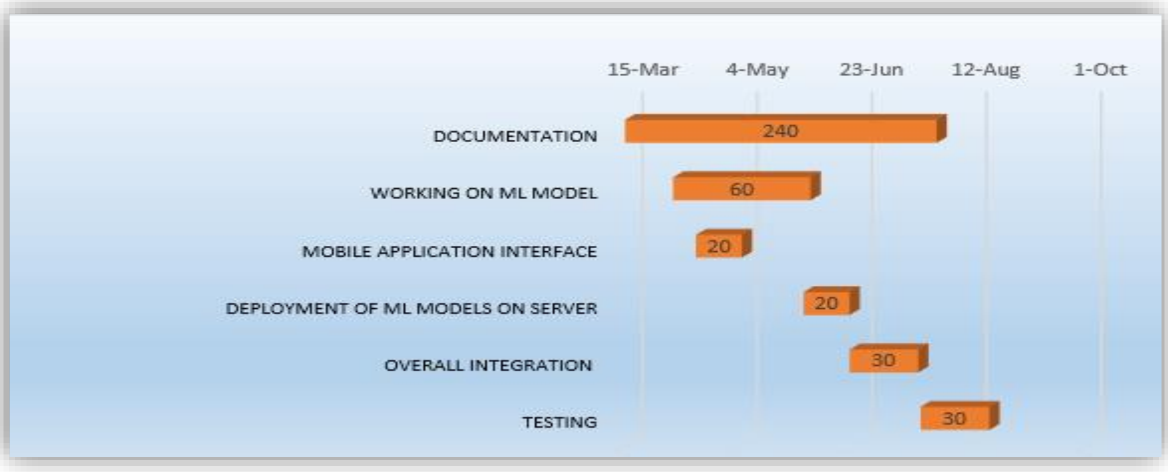
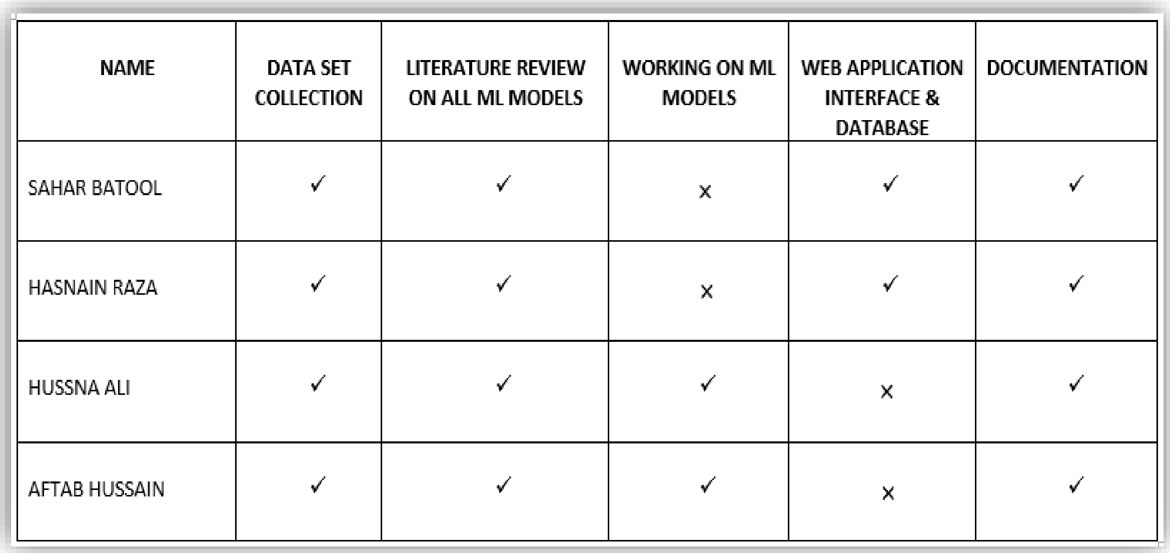


Figure 3 GANTT CHART (PHASE-II)

3.2.1 WORK DISTRIBUTION CHART

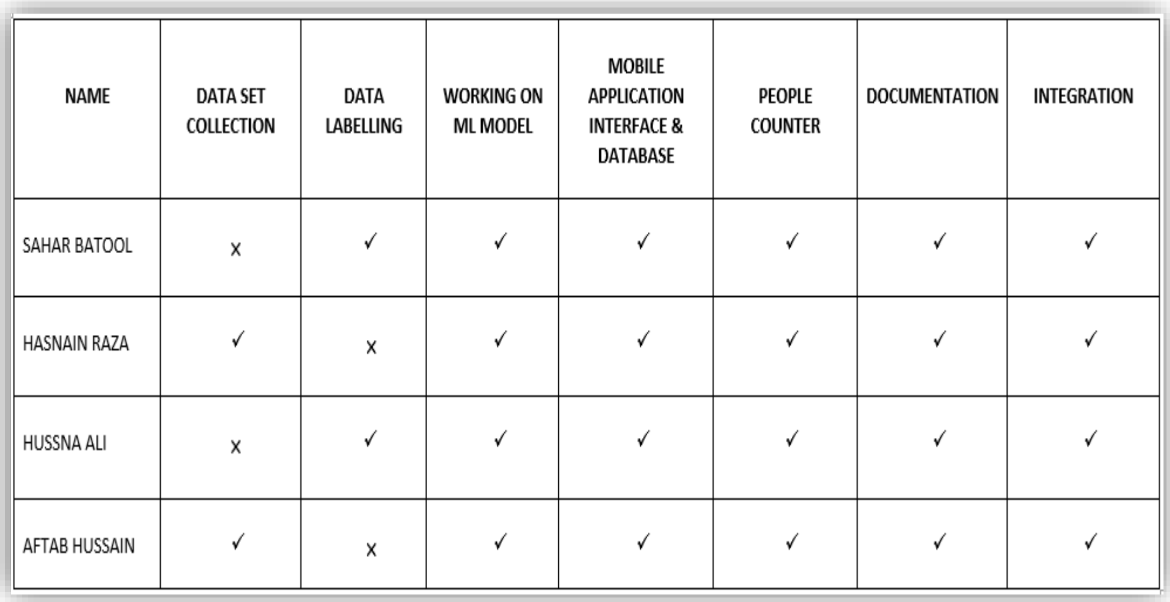
- **PHASE-I**



NAME	DATA SET COLLECTION	LITERATURE REVIEW ON ALL ML MODELS	WORKING ON ML MODELS	WEB APPLICATION INTERFACE & DATABASE	DOCUMENTATION
SAHAR BATOOL	✓	✓	x	✓	✓
HASNAIN RAZA	✓	✓	x	✓	✓
HUSSNA ALI	✓	✓	✓	x	✓
AFTAB HUSSAIN	✓	✓	✓	x	✓

Figure 4 WORKLOAD (PHASE-I)

- **PHASE-II**



NAME	DATA SET COLLECTION	DATA LABELLING	WORKING ON ML MODEL	MOBILE APPLICATION INTERFACE & DATABASE	PEOPLE COUNTER	DOCUMENTATION	INTEGRATION
SAHAR BATOOL	x	✓	✓	✓	✓	✓	✓
HASNAIN RAZA	✓	x	✓	✓	✓	✓	✓
HUSSNA ALI	x	✓	✓	✓	✓	✓	✓
AFTAB HUSSAIN	✓	x	✓	✓	✓	✓	✓

Figure 5 WORKLOAD (PHASE-II)

3.3 DATA GATHERING

What we did are as follows.

1. Firstly, we collected data sets for Weapon & Mask Detection from different sites.
2. As data sets were collected from different sites, we required to assemble them.
3. We refined the data sets by omitting all the blurred & irrelevant data.

3.4 EXPLORING ML MODELS

The steps are as follows.

1. We went through different research papers and found a suitable model for our project i.e.,

i. Sequential R-CNN (Mask Detection)

ii. YOLOv3 (Weapon Detection)

3.5 SOFTWARE

3.5.1 ANACONDA

We used Anaconda to train & test our ML models i.e., Weapon & Face Mask, and People Counter's code.

1. Weapon Detection Model; YOLOv3 ML model.
2. Face Mask Detection Model; Sequential R-CNN using libraries like: OpenCV, Keras, Numpy etc.
3. People Counting Code.

3.5.2 VISUAL LITE

We designed our Web App using Visual Lite i.e., Front-End & Back-End.

1. Front-End; HTML and CSS code.
2. Back-End; PHP Code & MySQL table.

3.5.3 MY SQL

We used MY SQL for creating data entry tables for Registration & Login pages and connected the database with interfaces via PHP.

3.5.4 ANDROID STUDIO

We designed our Mobile App using Android Studio i.e., Front-End & Back-End.

1. Front-End; For designing the application's interface via JAVA & XML files.
2. Back-End; For connecting to database via SQLITE.

CHAPTER: 4 DESIGNING AND DEVELOPMENT

4.1 DESIGN

Our project design consists of following diagrams.

4.1.1 FLOW DIAGRAM

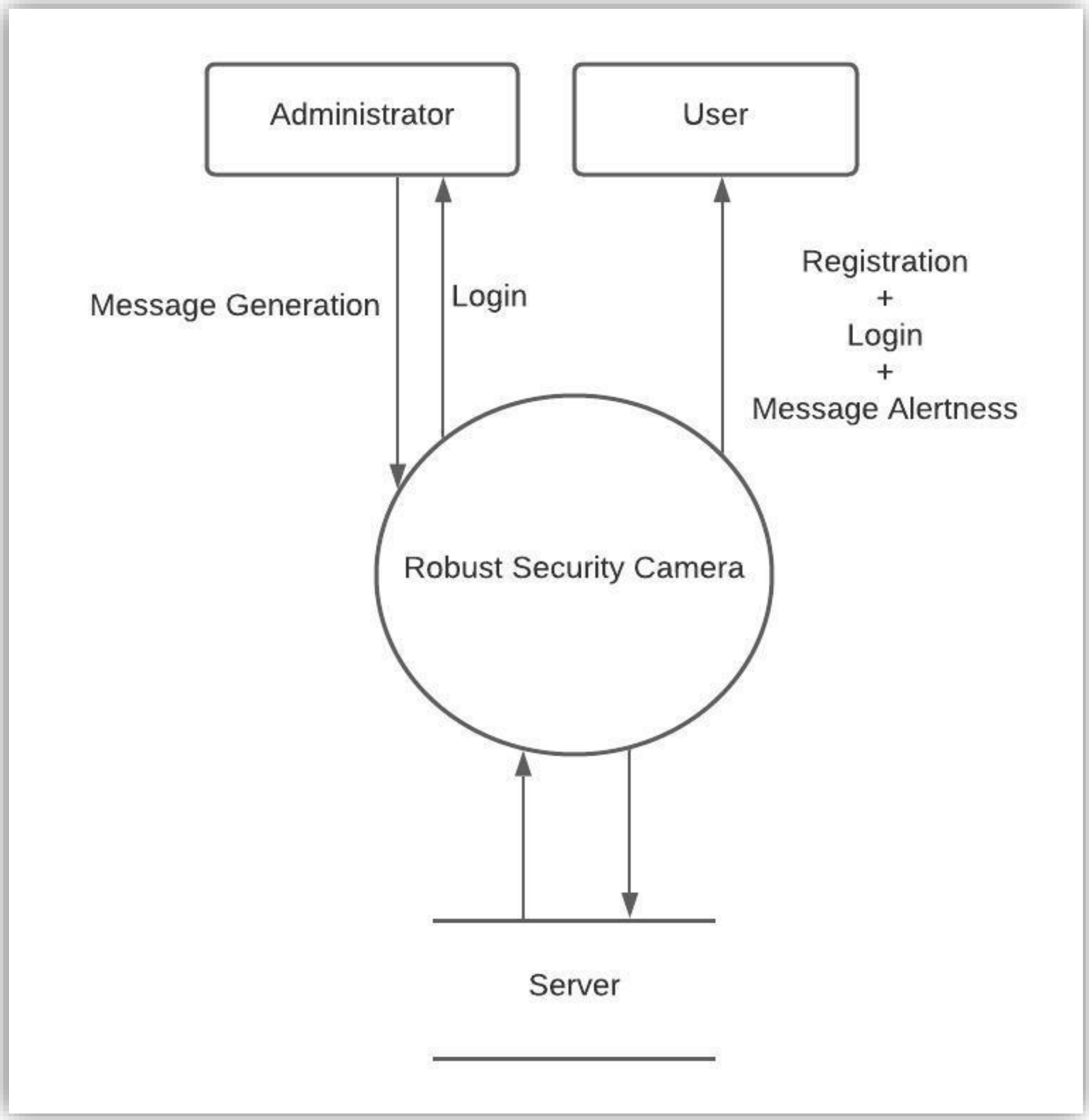


Figure 6 FLOW DIAGRAM

4.1.2 USE CASE DIAGRAM

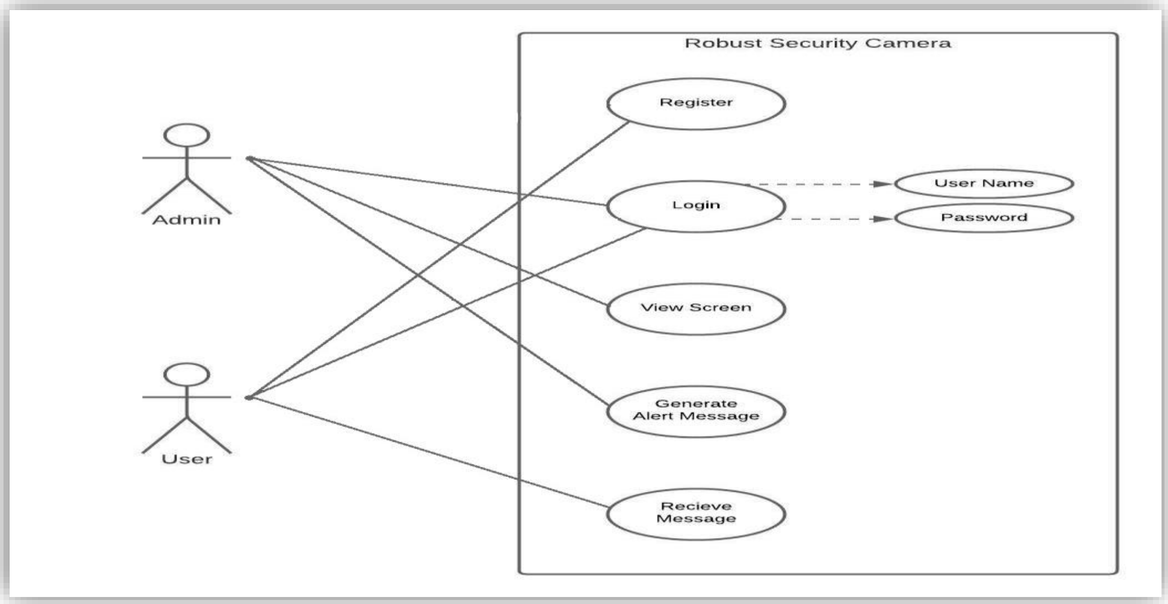


Figure 7 USE CASE DIAGRAM

5 ER DIAGRAM

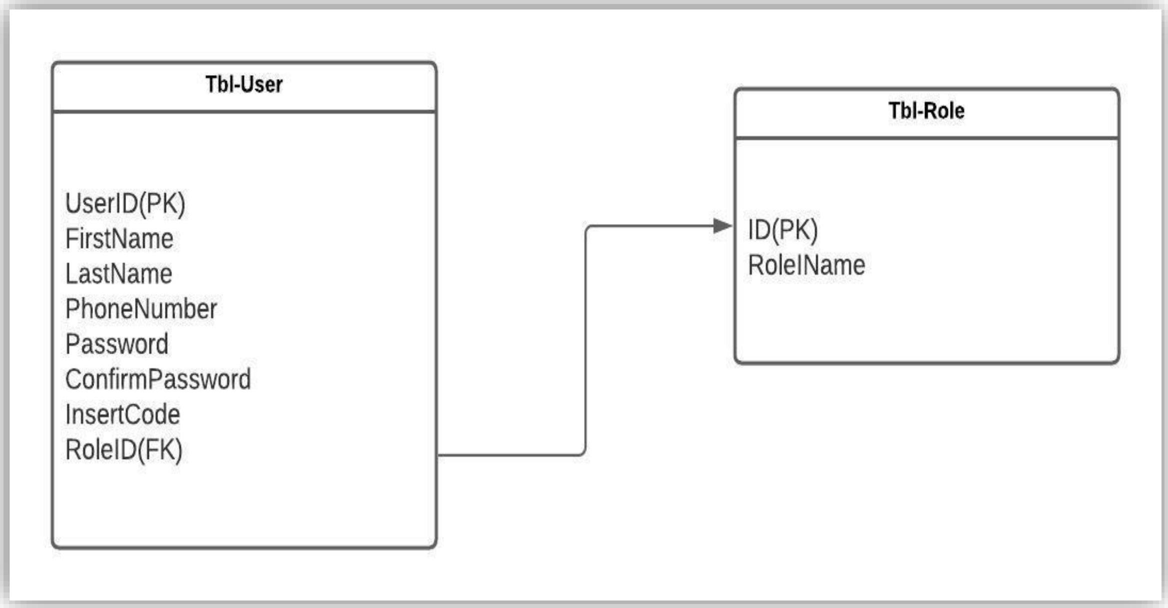


Figure 8 ER DIAGRAM

4.1.3 ACTIVITY DIAGRAM (USER SIDE)

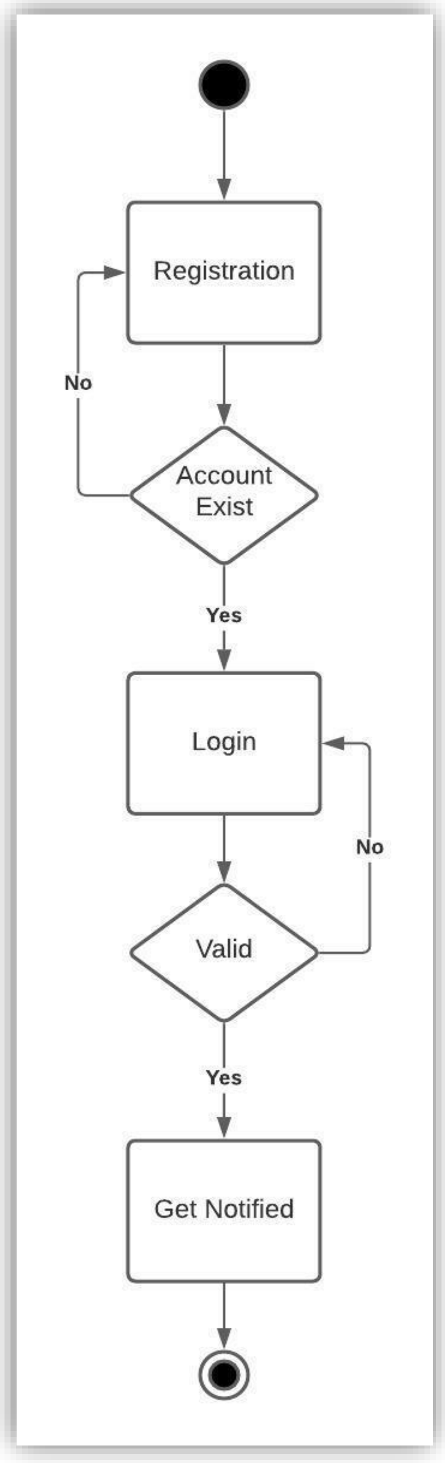


Figure 9 ACTIVITY DIAGRAM(USER)

4.1.4 ACTIVITY DIAGRAM (ADMIN SIDE)

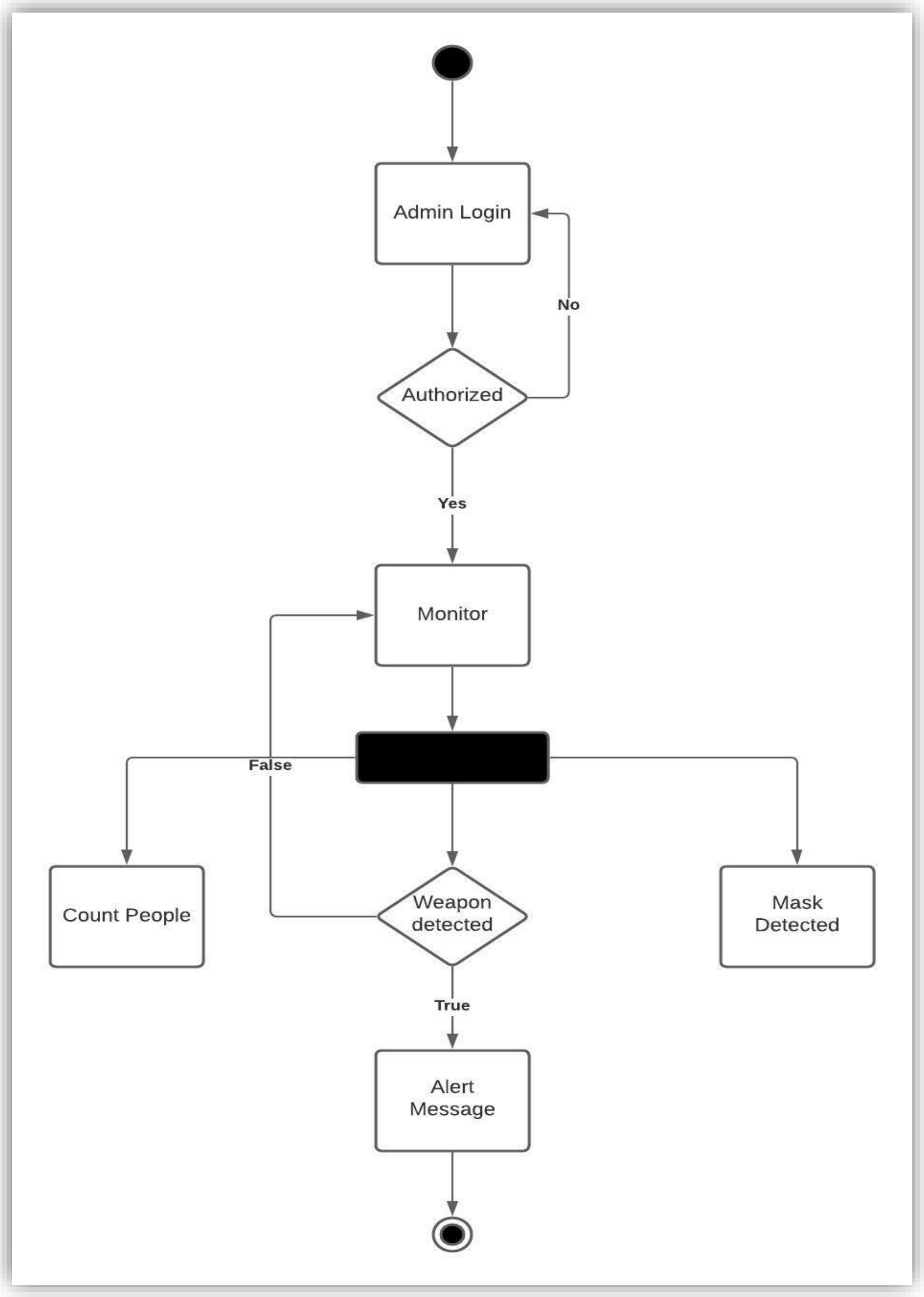


Figure 10 ACTIVITY DIAGRAM(ADMIN)

4.1.5 SEQUENCE DIAGRAM

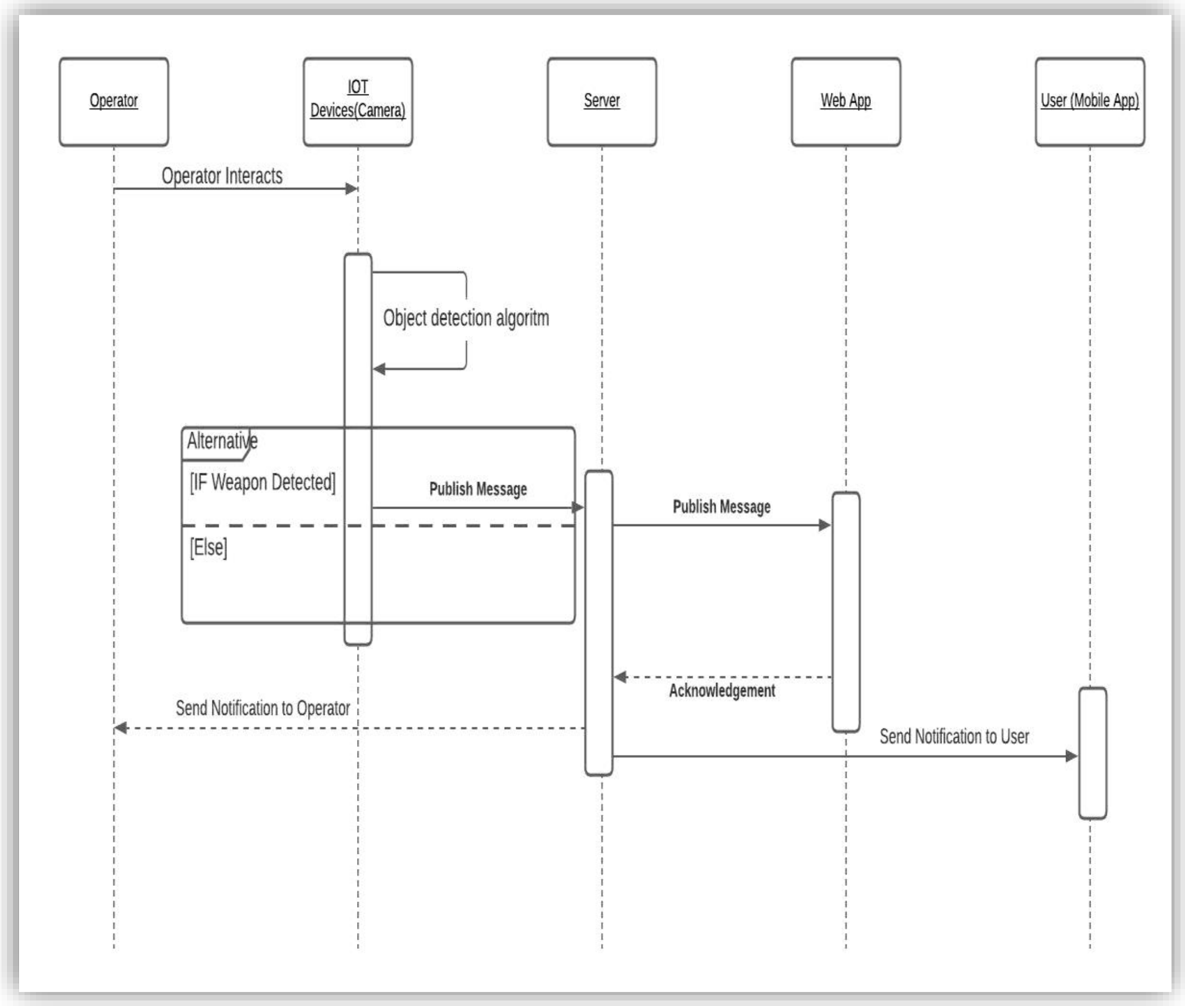


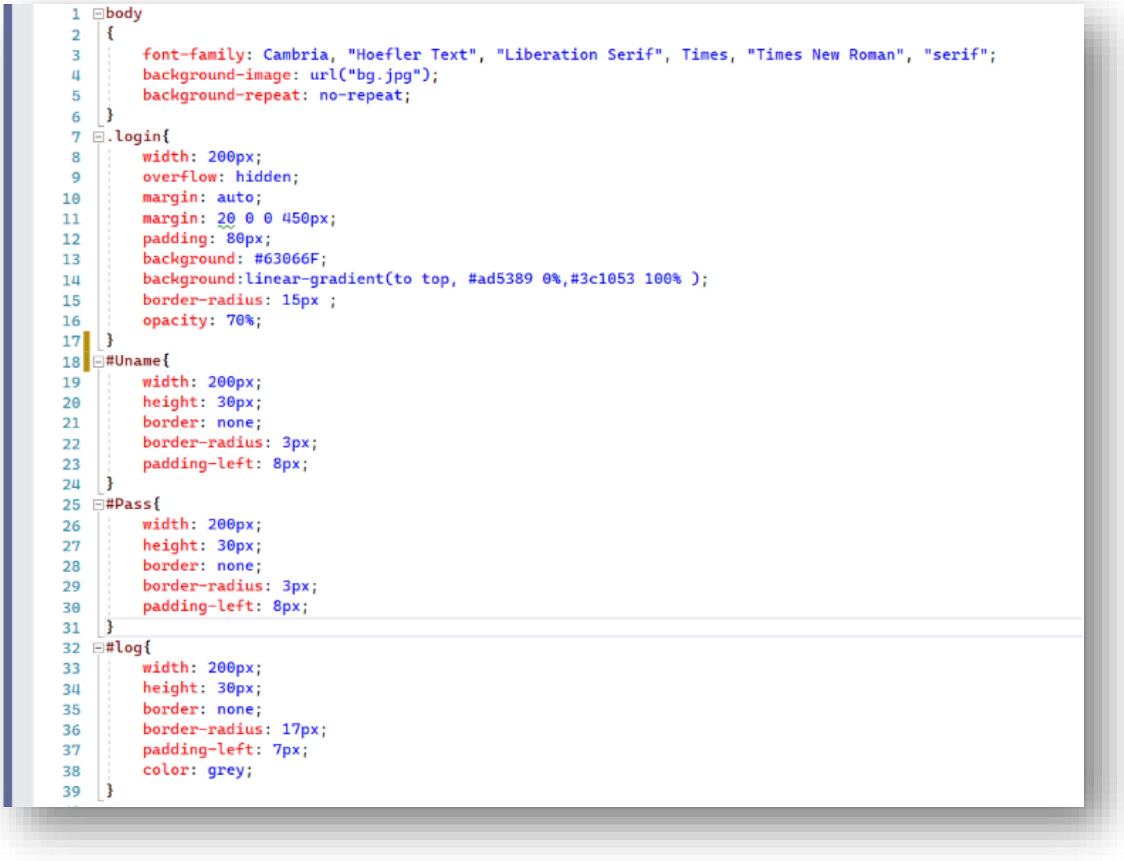
Figure 11 SEQUENCE DIAGRAM

4.2 DEVELOPMENT

Our project consists of both Web Application (to view screen and as well as automatically generate message when any weapon detected) and Mobile Application (to receive Alert Message). Thus, we developed our web application using Dreamweaver and Visual Studio (Dreamweaver for Html and CSS Code and Visual Studio for PHP Code) and we developed our Mobile Application on Android Studio.

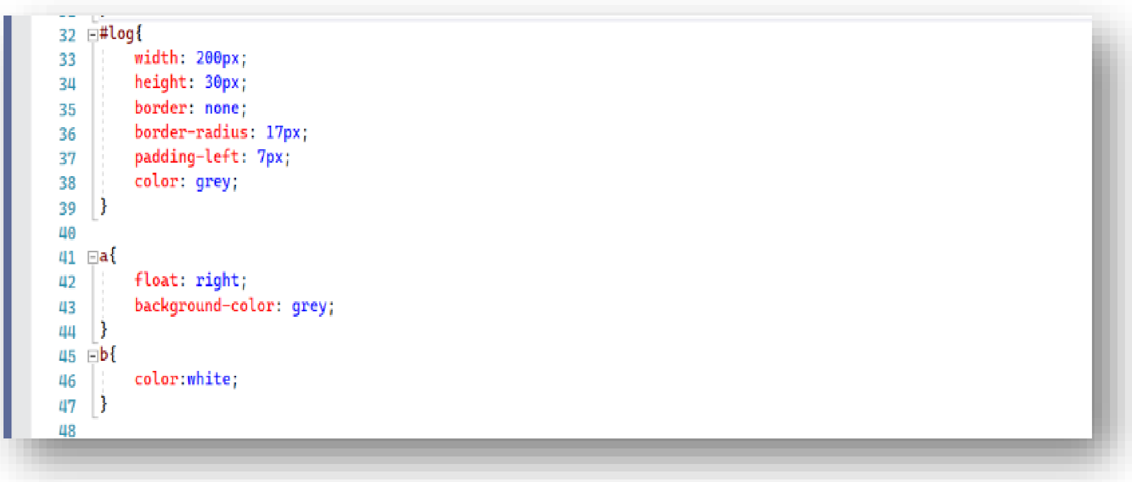
4.2.1 WEB APPLICATION’S CODE

4.2.1.1 CSS CODE FOR LOGIN (WEBSITE)



```css
body
{
    font-family: Cambria, "Hoefler Text", "Liberation Serif", Times, "Times New Roman", "serif";
    background-image: url("bg.jpg");
    background-repeat: no-repeat;
}
.login{
    width: 200px;
    overflow: hidden;
    margin: auto;
    margin: 20 0 0 450px;
    padding: 80px;
    background: #63066F;
    background:linear-gradient(to top, #ad5389 0%,#3c1053 100% );
    border-radius: 15px ;
    opacity: 70%;
}
#Uname{
    width: 200px;
    height: 30px;
    border: none;
    border-radius: 3px;
    padding-left: 8px;
}
#Pass{
    width: 200px;
    height: 30px;
    border: none;
    border-radius: 3px;
    padding-left: 8px;
}
#log{
    width: 200px;
    height: 30px;
    border: none;
    border-radius: 17px;
    padding-left: 7px;
    color: grey;
}
```

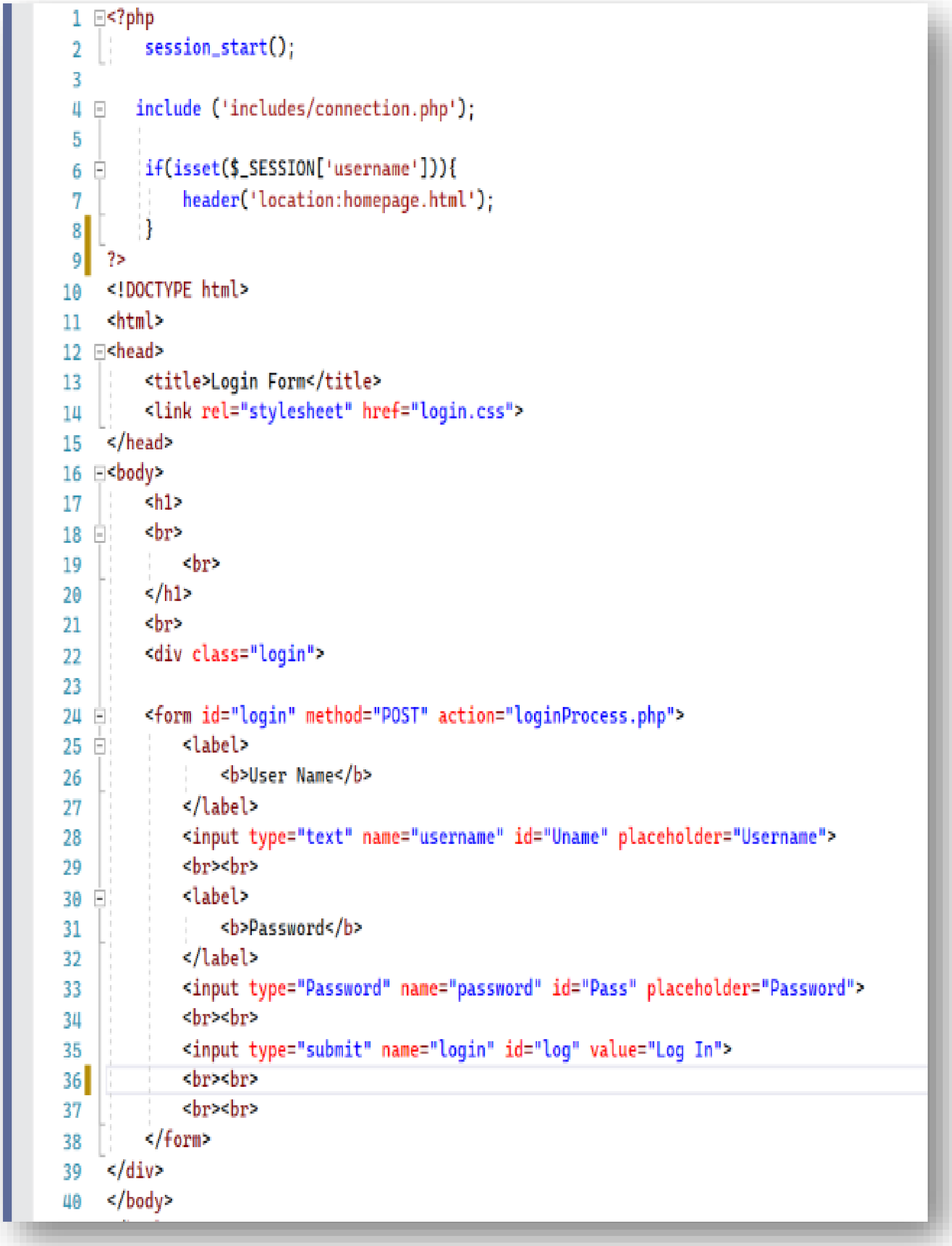
Figure 12 CSS CODE FOR LOGIN (WEBSITE) (1)



```css
#log{
    width: 200px;
    height: 30px;
    border: none;
    border-radius: 17px;
    padding-left: 7px;
    color: grey;
}

a{
    float: right;
    background-color: grey;
}
b{
    color:white;
}
```

Figure 13 CSS CODE FOR LOGIN (WEBSITE) (2)

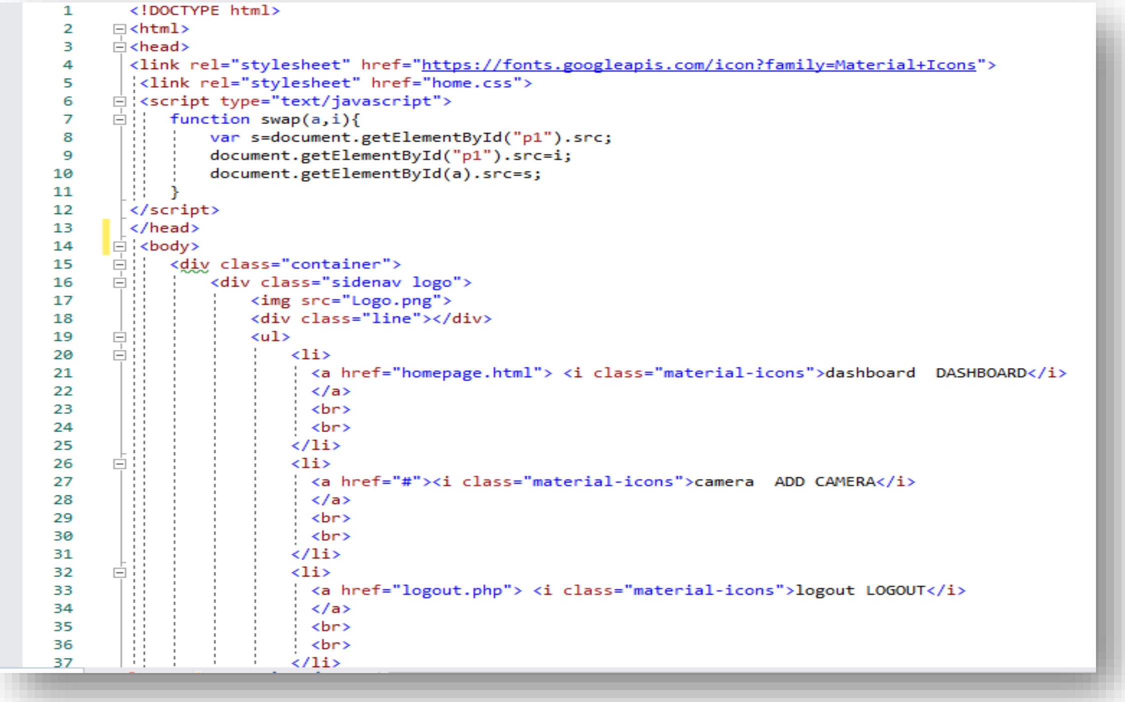
4.2.1.2 PHP CODE FOR LOGIN (WEBSITE)



```
<?php
    session_start();

   include ('includes/connection.php');

    if(isset($_SESSION['username'])){
        header('location:homepage.html');
    }
?>
<!DOCTYPE html>
<html>
<head>
    <title>Login Form</title>
    <link rel="stylesheet" href="login.css">
</head>
<body>
    <h1>
    <br>
        <br>
    </h1>
    <br>
    <div class="login">

    <form id="login" method="POST" action="loginProcess.php">
        <label>
            <b>User Name</b>
        </label>
        <input type="text" name="username" id="Uname" placeholder="Username">
        <br><br>
        <label>
            <b>Password</b>
        </label>
        <input type="Password" name="password" id="Pass" placeholder="Password">
        <br><br>
        <input type="submit" name="login" id="log" value="Log In">
        <br><br>
        <br><br>
    </form>
</div>
</body>
```

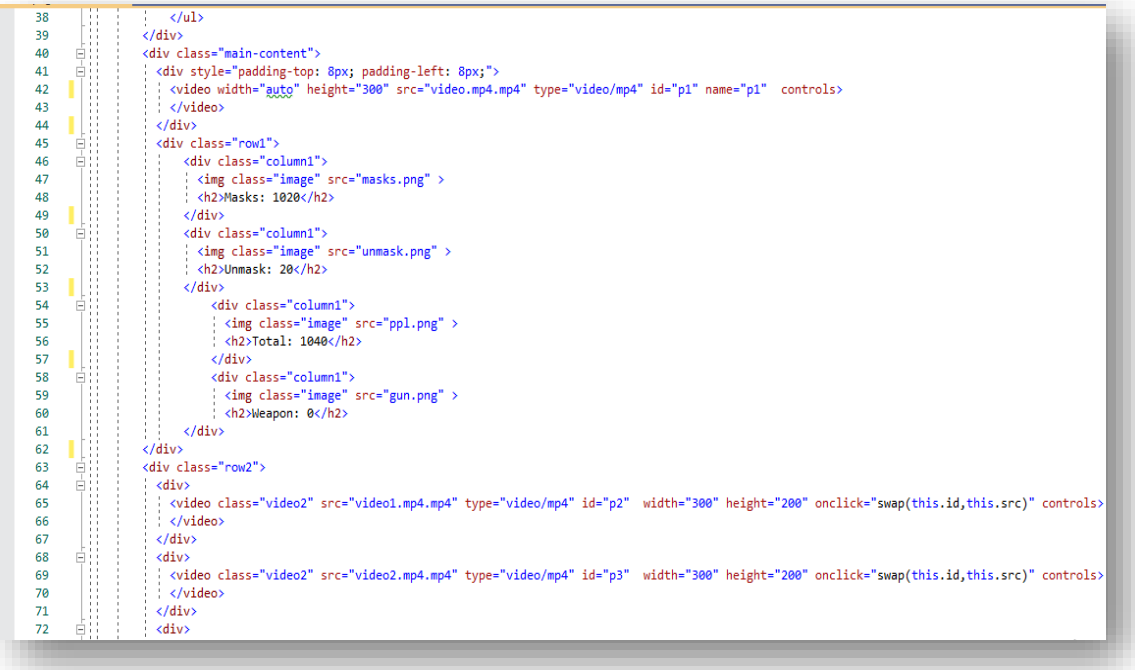
Figure 14 PHP CODE FOR LOGIN (WEBSITE)

4.2.1.3 HTML CODE FOR HOME(WEBSITE)



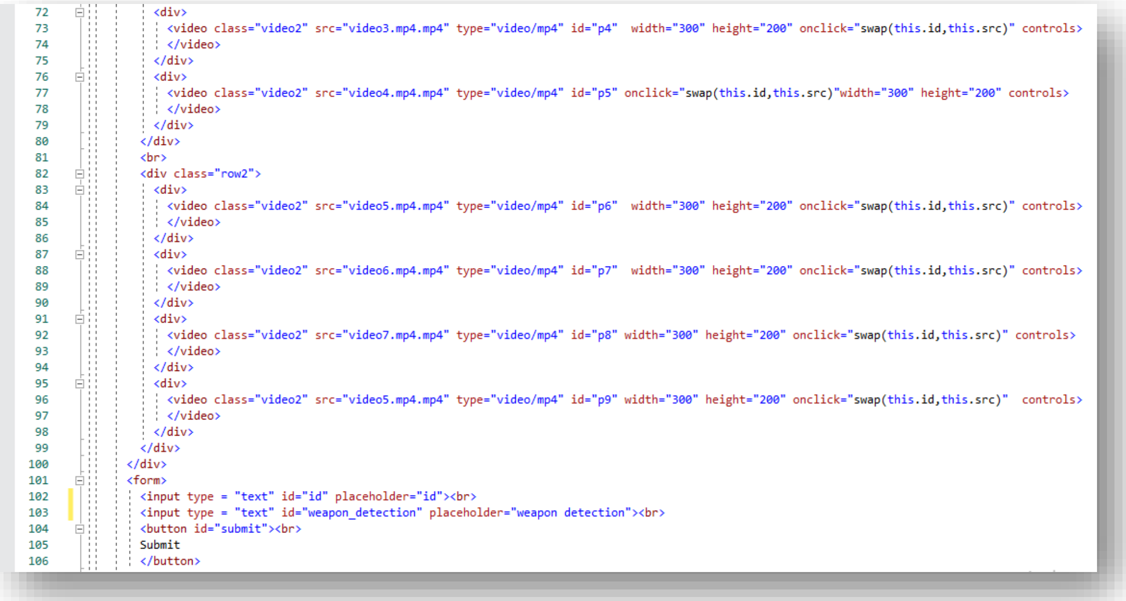
```html
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">
 <link rel="stylesheet" href="home.css">
 <script type="text/javascript">
    function swap(a,i){
        var s=document.getElementById("p1").src;
        document.getElementById("p1").src=i;
        document.getElementById(a).src=s;
    }
</script>
</head>
 <body>
    <div class="container">
        <div class="sidenav logo">
            <img src="Logo.png">
            <div class="line"></div>
            <ul>
                <li>
                  <a href="homepage.html"> <i class="material-icons">dashboard  DASHBOARD</i>
                  </a>
                  <br>
                  <br>
                </li>
                <li>
                  <a href="#"><i class="material-icons">camera  ADD CAMERA</i>
                  </a>
                  <br>
                  <br>
                </li>
                <li>
                  <a href="logout.php"> <i class="material-icons">logout LOGOUT</i>
                  </a>
                  <br>
                  <br>
                </li>
```

Figure 15 HTML CODE FOR HOME(WEBSITE) (1)



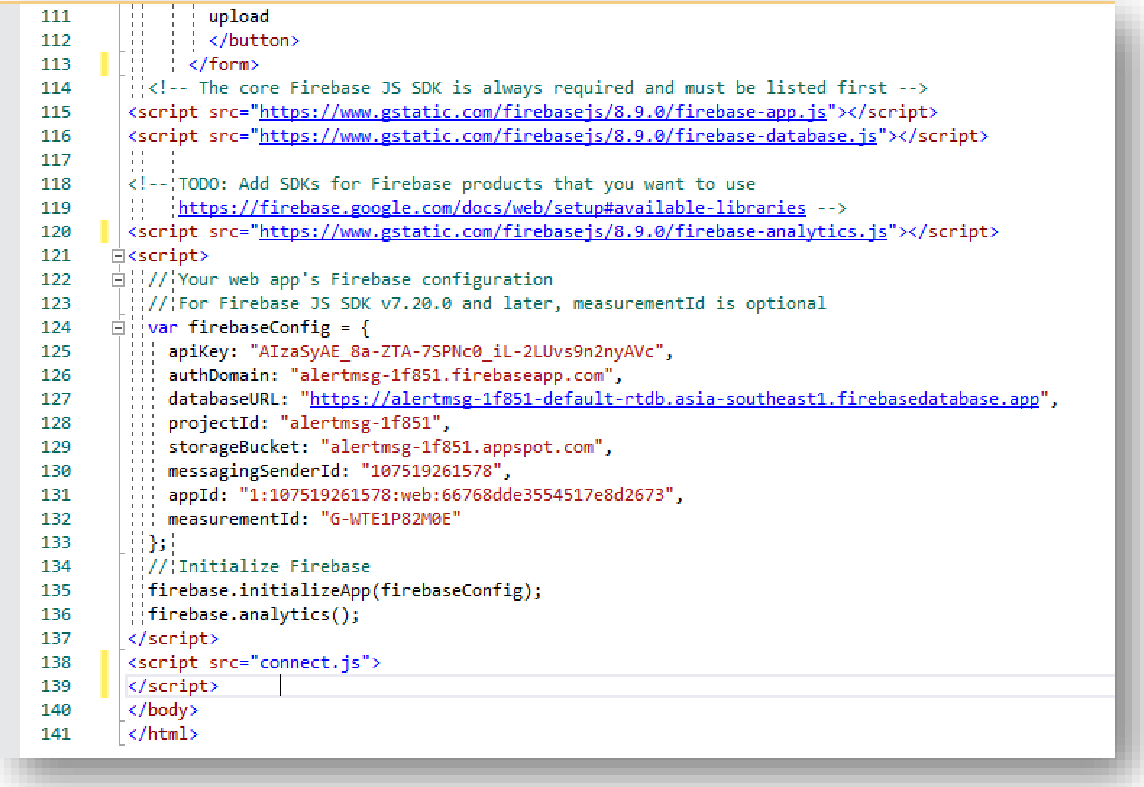
```html
        </ul>
    </div>
    <div class="main-content">
      <div style="padding-top: 8px; padding-left: 8px;">
        <video width="auto" height="300" src="video.mp4.mp4" type="video/mp4" id="p1" name="p1"  controls>
        </video>
      </div>
      <div class="row1">
          <div class="column1">
            <img class="image" src="masks.png" >
            <h2>Masks: 1020</h2>
          </div>
          <div class="column1">
            <img class="image" src="unmask.png" >
            <h2>Unmask: 20</h2>
          </div>
              <div class="column1">
                <img class="image" src="ppl.png" >
                <h2>Total: 1040</h2>
              </div>
              <div class="column1">
                <img class="image" src="gun.png" >
                <h2>Weapon: 0</h2>
          </div>
    </div>
    <div class="row2">
      <div>
        <video class="video2" src="video1.mp4.mp4" type="video/mp4" id="p2"  width="300" height="200" onclick="swap(this.id,this.src)" controls>
        </video>
      </div>
      <div>
        <video class="video2" src="video2.mp4.mp4" type="video/mp4" id="p3"  width="300" height="200" onclick="swap(this.id,this.src)" controls>
        </video>
      </div>
      <div>
```

Figure 16 HTML CODE FOR HOME(WEBSITE) (2)



```
        <div>
          <video class="video2" src="video3.mp4.mp4" type="video/mp4" id="p4"  width="300" height="200" onclick="swap(this.id,this.src)" controls>
          </video>
        </div>
        <div>
          <video class="video2" src="video4.mp4.mp4" type="video/mp4" id="p5" onclick="swap(this.id,this.src)"width="300" height="200" controls>
          </video>
        </div>
      </div>
      <br>
      <div class="row2">
        <div>
          <video class="video2" src="video5.mp4.mp4" type="video/mp4" id="p6"  width="300" height="200" onclick="swap(this.id,this.src)" controls>
          </video>
        </div>
        <div>
          <video class="video2" src="video6.mp4.mp4" type="video/mp4" id="p7"  width="300" height="200" onclick="swap(this.id,this.src)" controls>
          </video>
        </div>
        <div>
          <video class="video2" src="video7.mp4.mp4" type="video/mp4" id="p8" width="300" height="200" onclick="swap(this.id,this.src)" controls>
          </video>
        </div>
        <div>
          <video class="video2" src="video5.mp4.mp4" type="video/mp4" id="p9" width="300" height="200" onclick="swap(this.id,this.src)"  controls>
          </video>
        </div>
      </div>
    </div>
    <form>
      <input type = "text" id="id" placeholder="id"><br>
      <input type = "text" id="weapon_detection" placeholder="weapon detection"><br>
      <button id="submit"><br>
      Submit
      </button>
```

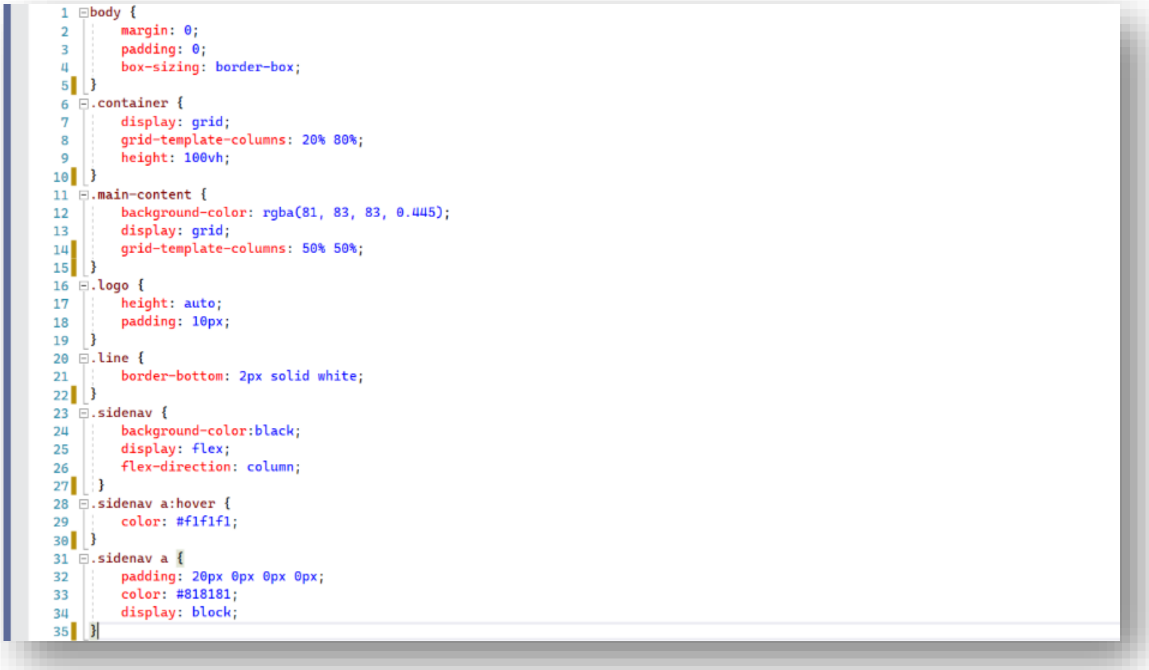
Figure 17 HTML CODE FOR HOME(WEBSITE) (3)



```
          upload
          </button>
        </form>
  <!-- The core Firebase JS SDK is always required and must be listed first -->
<script src="https://www.gstatic.com/firebasejs/8.9.0/firebase-app.js"></script>
<script src="https://www.gstatic.com/firebasejs/8.9.0/firebase-database.js"></script>

<!-- TODO: Add SDKs for Firebase products that you want to use
     https://firebase.google.com/docs/web/setup#available-libraries -->
<script src="https://www.gstatic.com/firebasejs/8.9.0/firebase-analytics.js"></script>
<script>
  // Your web app's Firebase configuration
  // For Firebase JS SDK v7.20.0 and later, measurementId is optional
  var firebaseConfig = {
    apiKey: "AIzaSyAE_8a-ZTA-7SPNc0_iL-2LUvs9n2nyAVc",
    authDomain: "alertmsg-1f851.firebaseapp.com",
    databaseURL: "https://alertmsg-1f851-default-rtdb.asia-southeast1.firebasedatabase.app",
    projectId: "alertmsg-1f851",
    storageBucket: "alertmsg-1f851.appspot.com",
    messagingSenderId: "107519261578",
    appId: "1:107519261578:web:66768dde3554517e8d2673",
    measurementId: "G-WTE1P82M0E"
  };
  // Initialize Firebase
  firebase.initializeApp(firebaseConfig);
  firebase.analytics();
</script>
<script src="connect.js">
</script>
</body>
</html>
```

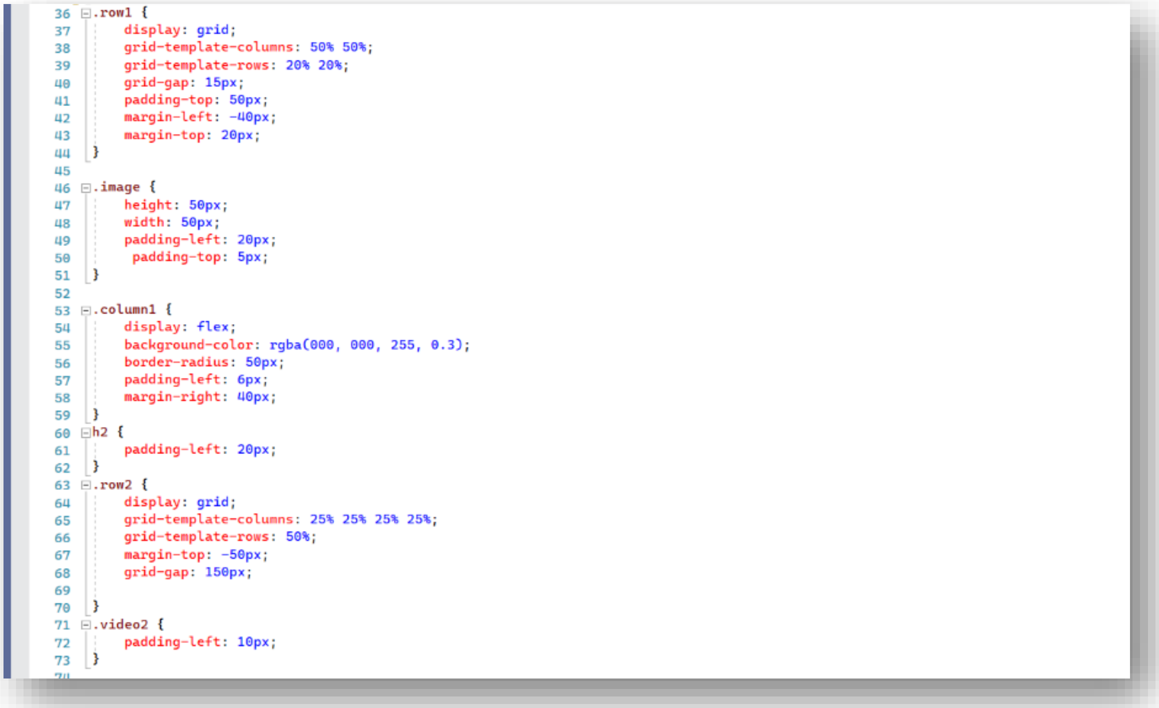
Figure 18 HTML CODE FOR HOME(WEBSITE) (4)

4.2.1.4 CSS CODE FOR HOME (WEBSITE)



```
body {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}
.container {
    display: grid;
    grid-template-columns: 20% 80%;
    height: 100vh;
}
.main-content {
    background-color: rgba(81, 83, 83, 0.445);
    display: grid;
    grid-template-columns: 50% 50%;
}
.logo {
    height: auto;
    padding: 10px;
}
.line {
    border-bottom: 2px solid white;
}
.sidenav {
    background-color:black;
    display: flex;
    flex-direction: column;
 }
.sidenav a:hover {
    color: #f1f1f1;
}
.sidenav a {
    padding: 20px 0px 0px 0px;
    color: #818181;
    display: block;
}
```

Figure 19 CSS CODE FOR HOME (WEBSITE) (1)



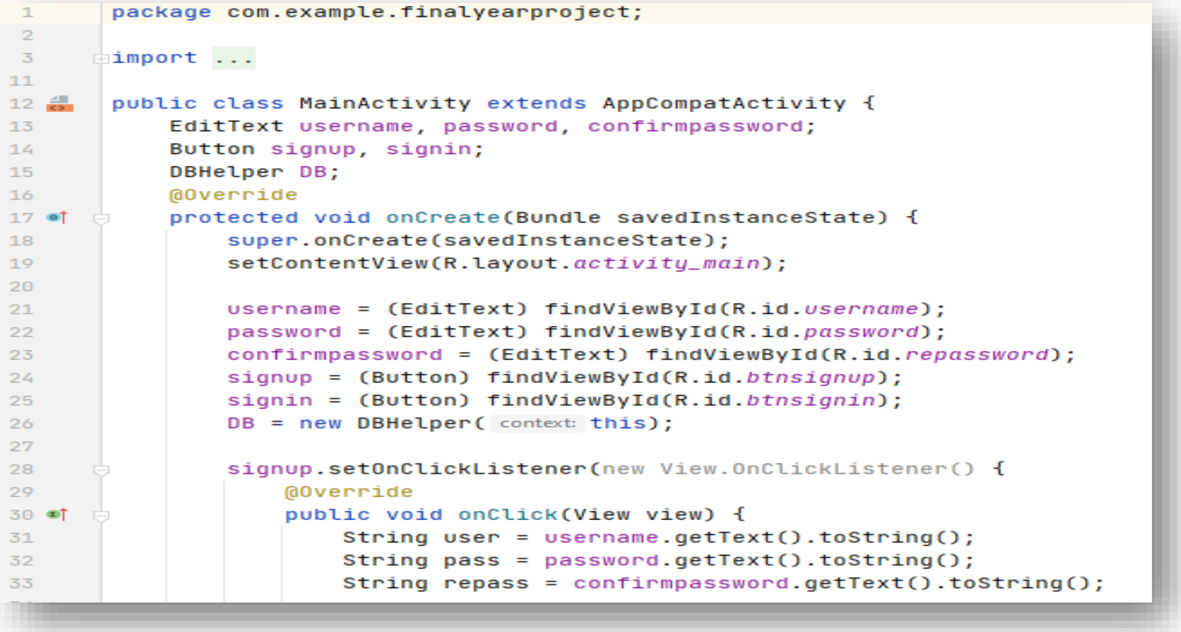
```
.row1 {
    display: grid;
    grid-template-columns: 50% 50%;
    grid-template-rows: 20% 20%;
    grid-gap: 15px;
    padding-top: 50px;
    margin-left: -40px;
    margin-top: 20px;
}

.image {
    height: 50px;
    width: 50px;
    padding-left: 20px;
     padding-top: 5px;
}

.column1 {
    display: flex;
    background-color: rgba(000, 000, 255, 0.3);
    border-radius: 50px;
    padding-left: 6px;
    margin-right: 40px;
}
h2 {
    padding-left: 20px;
}
.row2 {
    display: grid;
    grid-template-columns: 25% 25% 25% 25%;
    grid-template-rows: 50%;
    margin-top: -50px;
    grid-gap: 150px;

}
.video2 {
    padding-left: 10px;
}
```

Figure 20 CSS CODE FOR HOME (WEBSITE) (2)

4.2.2 MOBILE APPLICATION’S CODE

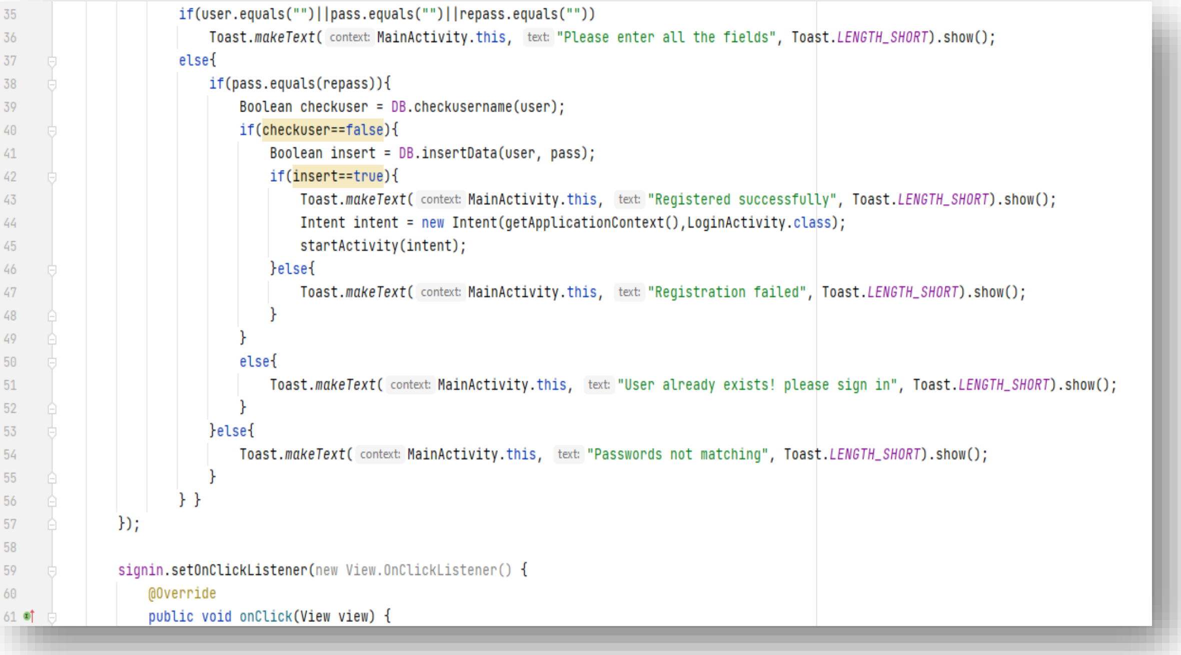
4.2.2.1 JAVA CODES

4.2.2.1.1 FOR REGISTRATION



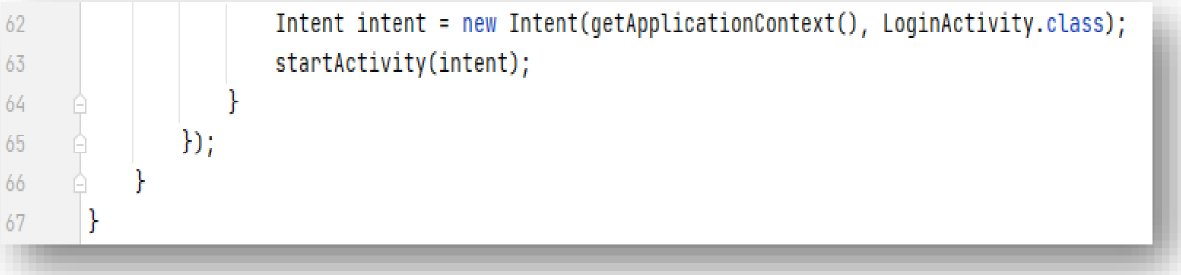
```java
package com.example.finalyearproject;

import ...

public class MainActivity extends AppCompatActivity {
    EditText username, password, confirmpassword;
    Button signup, signin;
    DBHelper DB;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        username = (EditText) findViewById(R.id.username);
        password = (EditText) findViewById(R.id.password);
        confirmpassword = (EditText) findViewById(R.id.repassword);
        signup = (Button) findViewById(R.id.btnsignup);
        signin = (Button) findViewById(R.id.btnsignin);
        DB = new DBHelper( context: this);

        signup.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String user = username.getText().toString();
                String pass = password.getText().toString();
                String repass = confirmpassword.getText().toString();
```

Figure 21 JAVA CODE FOR REGISTRATION (1)



```java
                if(user.equals("")||pass.equals("")||repass.equals(""))
                    Toast.makeText( context: MainActivity.this,  text: "Please enter all the fields", Toast.LENGTH_SHORT).show();
                else{
                    if(pass.equals(repass)){
                        Boolean checkuser = DB.checkusername(user);
                        if(checkuser==false){
                            Boolean insert = DB.insertData(user, pass);
                            if(insert==true){
                                Toast.makeText( context: MainActivity.this,  text: "Registered successfully", Toast.LENGTH_SHORT).show();
                                Intent intent = new Intent(getApplicationContext(),LoginActivity.class);
                                startActivity(intent);
                            }else{
                                Toast.makeText( context: MainActivity.this,  text: "Registration failed", Toast.LENGTH_SHORT).show();
                            }
                        }
                        else{
                            Toast.makeText( context: MainActivity.this,  text: "User already exists! please sign in", Toast.LENGTH_SHORT).show();
                        }
                    }else{
                        Toast.makeText( context: MainActivity.this,  text: "Passwords not matching", Toast.LENGTH_SHORT).show();
                    }
                } }
        });

        signin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
```

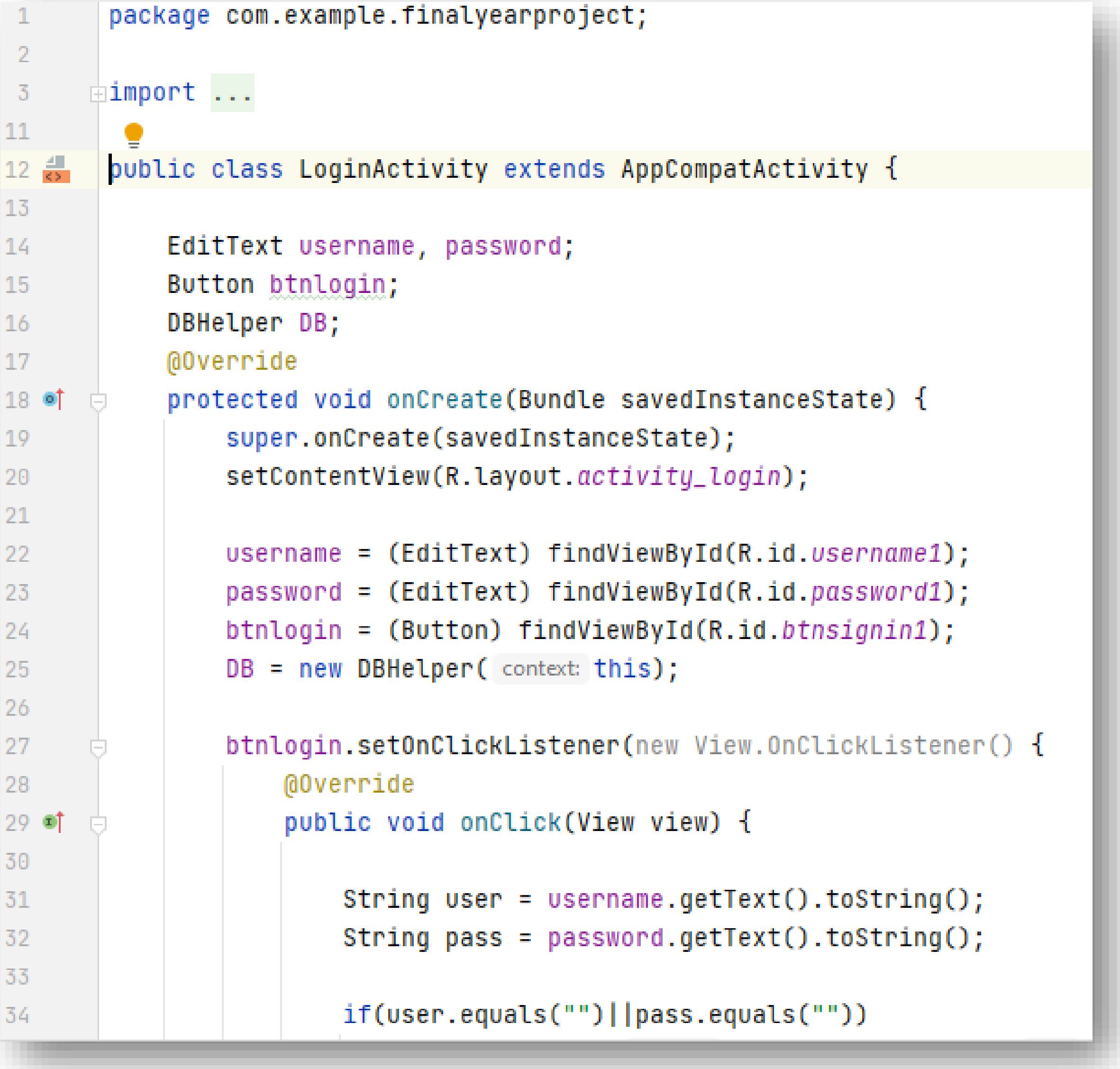
Figure 22 JAVA CODE FOR REGISTRATION (2)



```
                Intent intent = new Intent(getApplicationContext(), LoginActivity.class);
                startActivity(intent);
            }
        });
    }
}
```

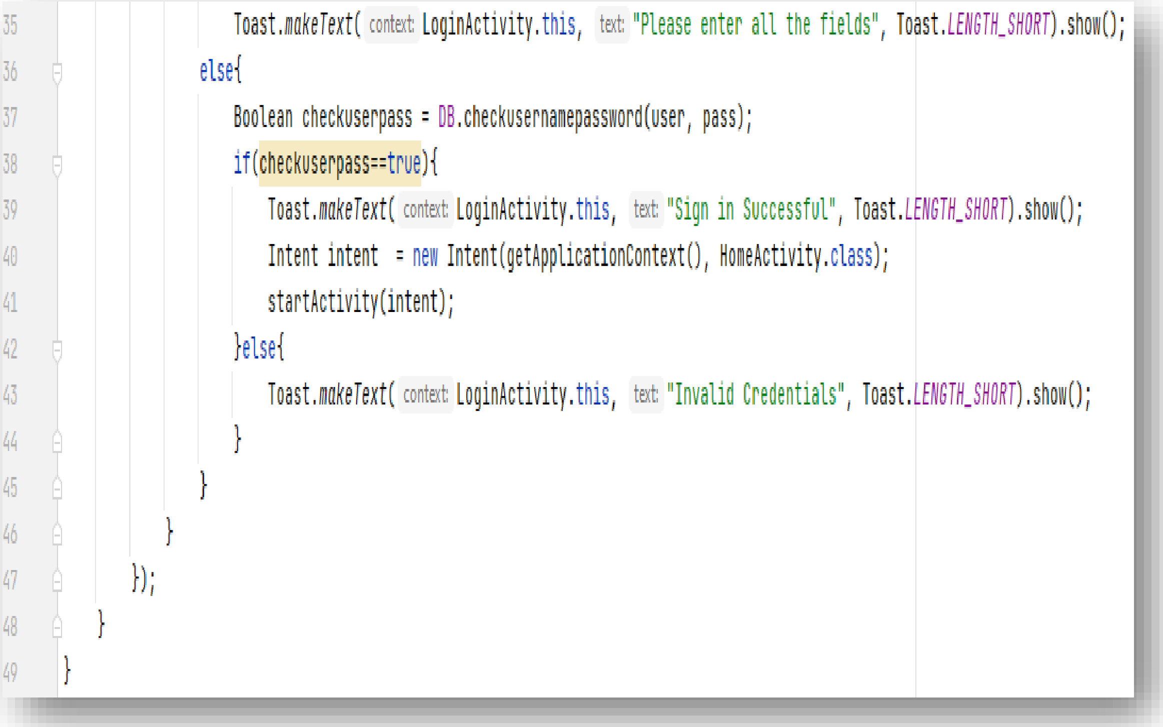
Figure 23 JAVA CODE FOR REGISTRATION (3)

4.2.2.1.2 FOR LOGIN



```
package com.example.finalyearproject;

import ...

public class LoginActivity extends AppCompatActivity {

    EditText username, password;
    Button btnlogin;
    DBHelper DB;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        username = (EditText) findViewById(R.id.username1);
        password = (EditText) findViewById(R.id.password1);
        btnlogin = (Button) findViewById(R.id.btnsignin1);
        DB = new DBHelper( context: this);

        btnlogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                String user = username.getText().toString();
                String pass = password.getText().toString();

                if(user.equals("")||pass.equals(""))
```

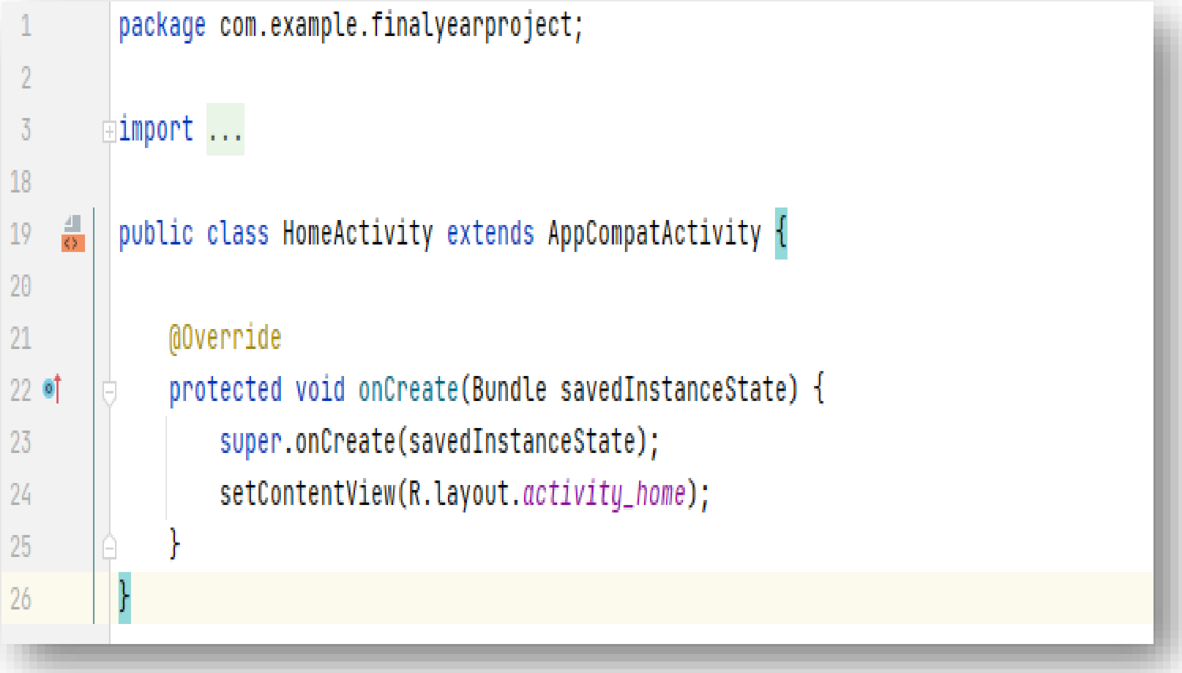
Figure 24 JAVA CODE FOR LOGIN (1)



```
                    Toast.makeText( context: LoginActivity.this,  text: "Please enter all the fields", Toast.LENGTH_SHORT).show();
                else{
                    Boolean checkuserpass = DB.checkusernamepassword(user, pass);
                    if(checkuserpass==true){
                        Toast.makeText( context: LoginActivity.this,  text: "Sign in Successful", Toast.LENGTH_SHORT).show();
                        Intent intent  = new Intent(getApplicationContext(), HomeActivity.class);
                        startActivity(intent);
                    }else{
                        Toast.makeText( context: LoginActivity.this,  text: "Invalid Credentials", Toast.LENGTH_SHORT).show();
                    }
                }
            }
        });
    }
}
```

Figure 25 JAVA CODE FOR LOGIN (2)

4.2.2.1.3 FOR HOME

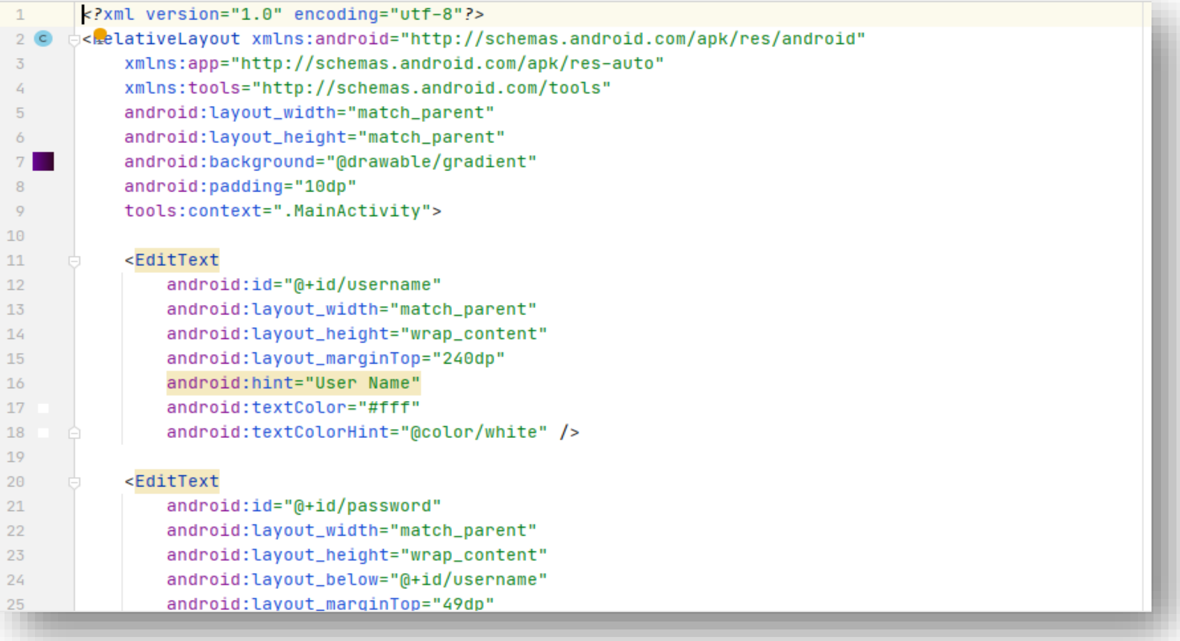


```
package com.example.finalyearproject;

import ...

public class HomeActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);
    }
}
```

Figure 26 JAVA CODE FOR HOME

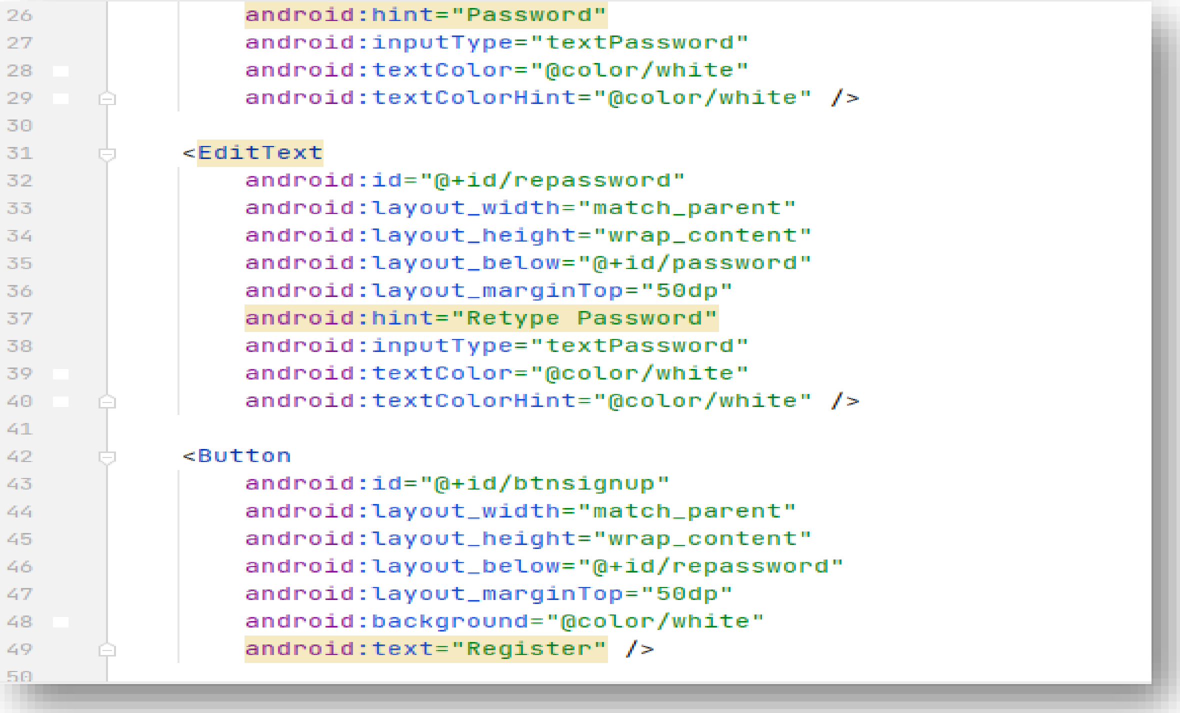
4.2.2.2 XML CODES

4.2.2.2.1 FOR REGISTRATION



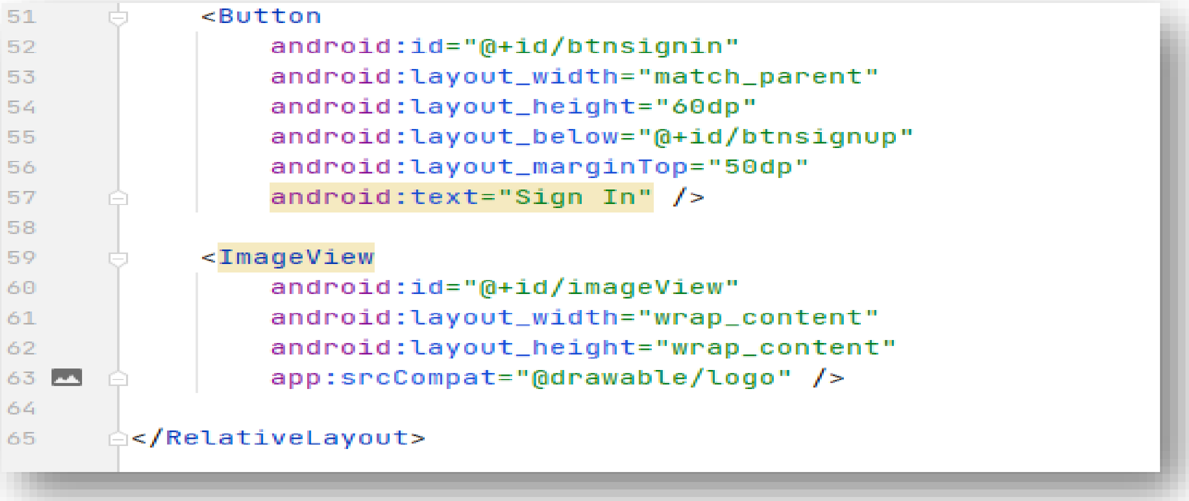
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/gradient"
    android:padding="10dp"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/username"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="240dp"
        android:hint="User Name"
        android:textColor="#fff"
        android:textColorHint="@color/white" />

    <EditText
        android:id="@+id/password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/username"
        android:layout_marginTop="49dp"
```

Figure 27 XML CODE FOR REGISTRATION (1)



```
        android:hint="Password"
        android:inputType="textPassword"
        android:textColor="@color/white"
        android:textColorHint="@color/white" />

    <EditText
        android:id="@+id/repassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/password"
        android:layout_marginTop="50dp"
        android:hint="Retype Password"
        android:inputType="textPassword"
        android:textColor="@color/white"
        android:textColorHint="@color/white" />

    <Button
        android:id="@+id/btnsignup"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/repassword"
        android:layout_marginTop="50dp"
        android:background="@color/white"
        android:text="Register" />
```

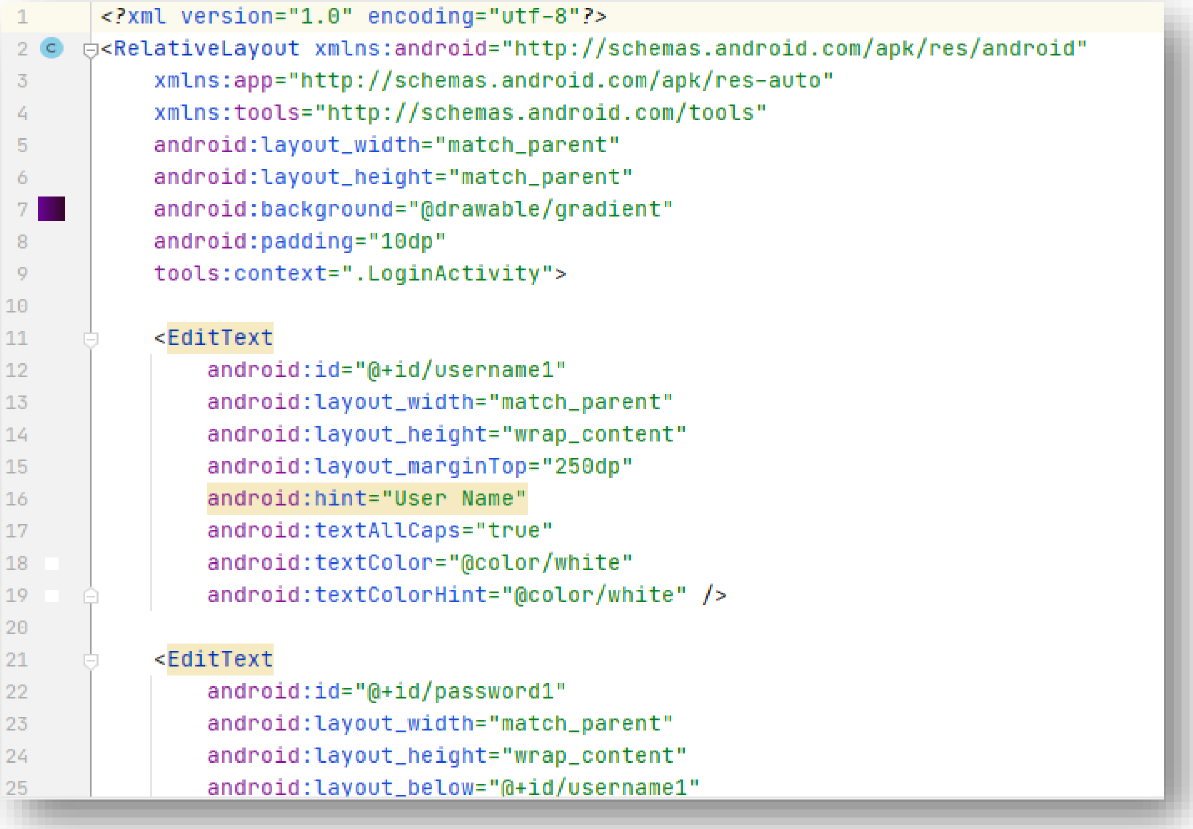
Figure 28 XML CODE FOR REGISTRATION (2)



```
    <Button
        android:id="@+id/btnsignin"
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:layout_below="@+id/btnsignup"
        android:layout_marginTop="50dp"
        android:text="Sign In" />

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:srcCompat="@drawable/logo" />

</RelativeLayout>
```

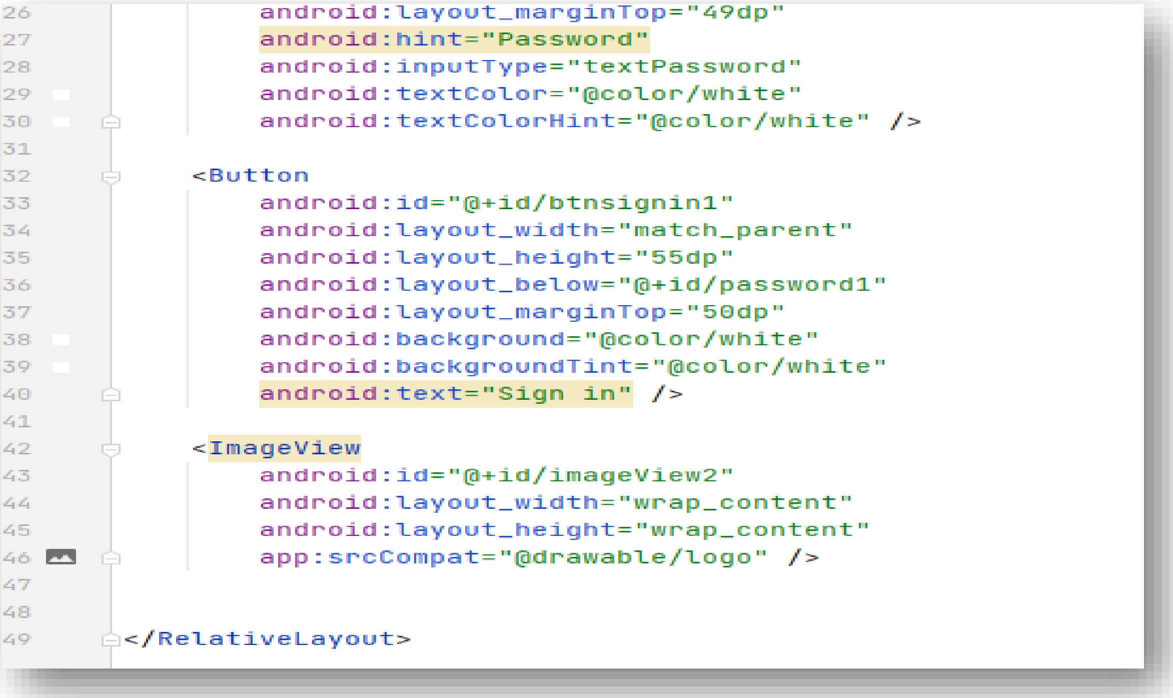
Figure 29 XML CODE FOR REGISTRATION (3)

4.2.2.2.2 FOR LOGIN



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/gradient"
    android:padding="10dp"
    tools:context=".LoginActivity">

    <EditText
        android:id="@+id/username1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="250dp"
        android:hint="User Name"
        android:textAllCaps="true"
        android:textColor="@color/white"
        android:textColorHint="@color/white" />

    <EditText
        android:id="@+id/password1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/username1"
```

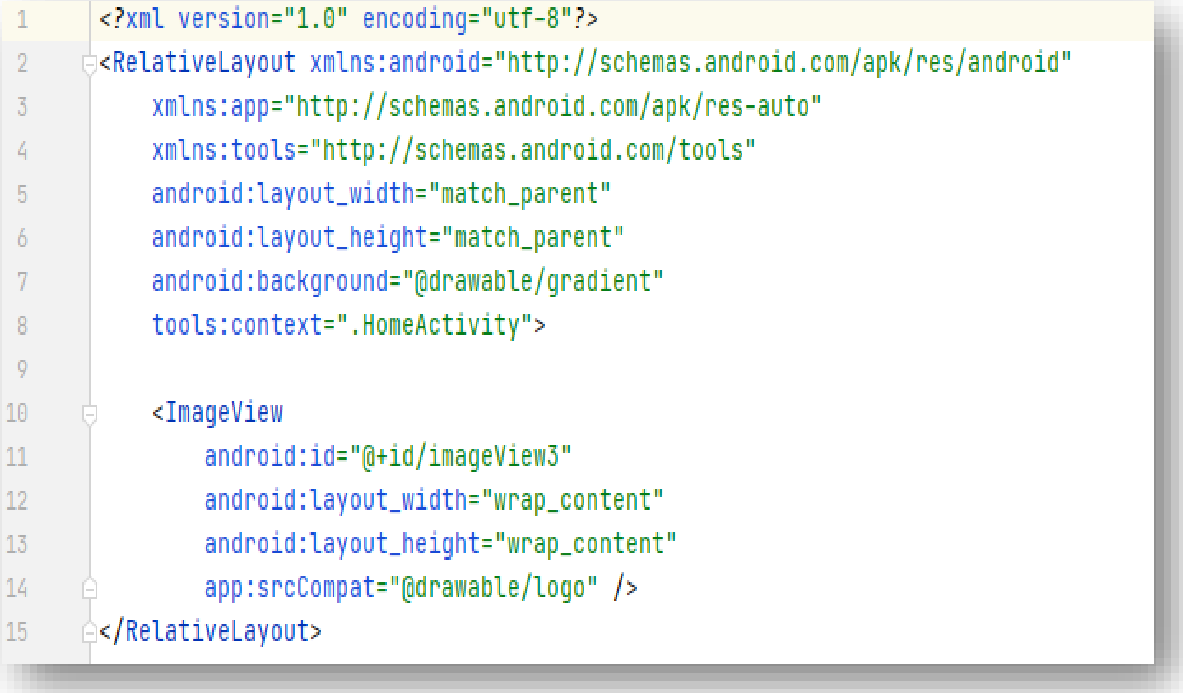
Figure 30 XML CODE FOR LOGIN (1)



```
            android:layout_marginTop="49dp"
            android:hint="Password"
            android:inputType="textPassword"
            android:textColor="@color/white"
            android:textColorHint="@color/white" />

        <Button
            android:id="@+id/btnsignin1"
            android:layout_width="match_parent"
            android:layout_height="55dp"
            android:layout_below="@+id/password1"
            android:layout_marginTop="50dp"
            android:background="@color/white"
            android:backgroundTint="@color/white"
            android:text="Sign in" />

        <ImageView
            android:id="@+id/imageView2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            app:srcCompat="@drawable/logo" />


</RelativeLayout>
```

Figure 31 XML CODE FOR LOGIN (2)

4.2.2.2.3 FOR HOME



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/gradient"
    tools:context=".HomeActivity">

    <ImageView
        android:id="@+id/imageView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:srcCompat="@drawable/logo" />
</RelativeLayout>
```

Figure 32 XML CODE FOR HOME

CHAPTER: 5 IMPLEMENTATIONS

5.1 WEB INTERFACE

5.1.1 LOGIN PAGE

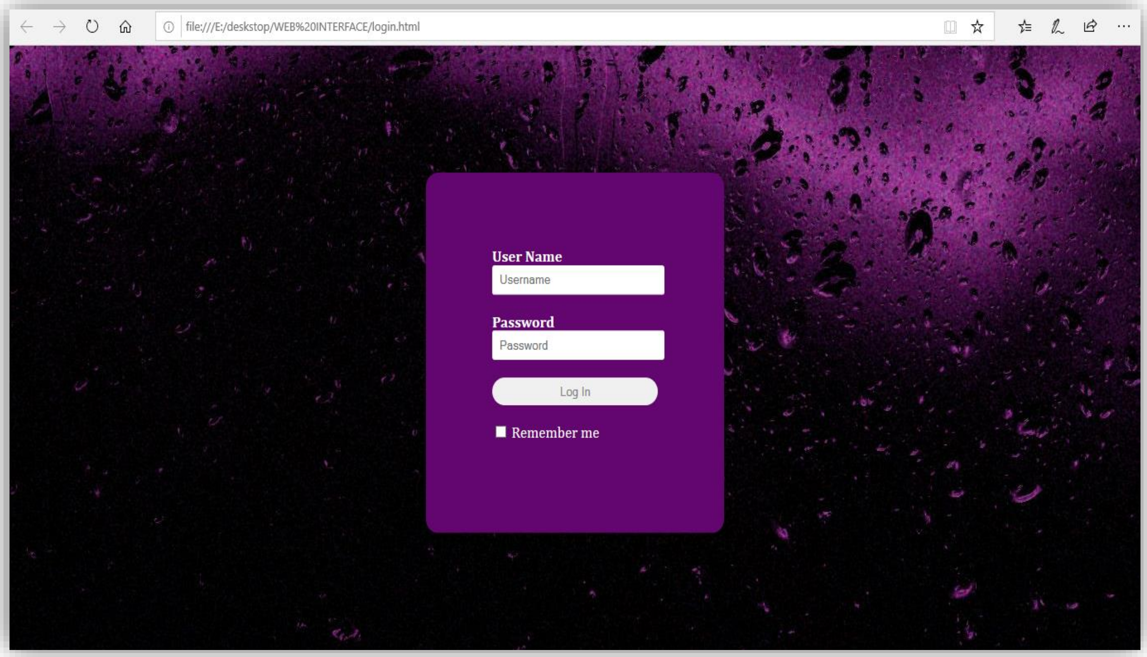


Figure 33 LOGIN PAGE

5.1.2 MAIN PAGE

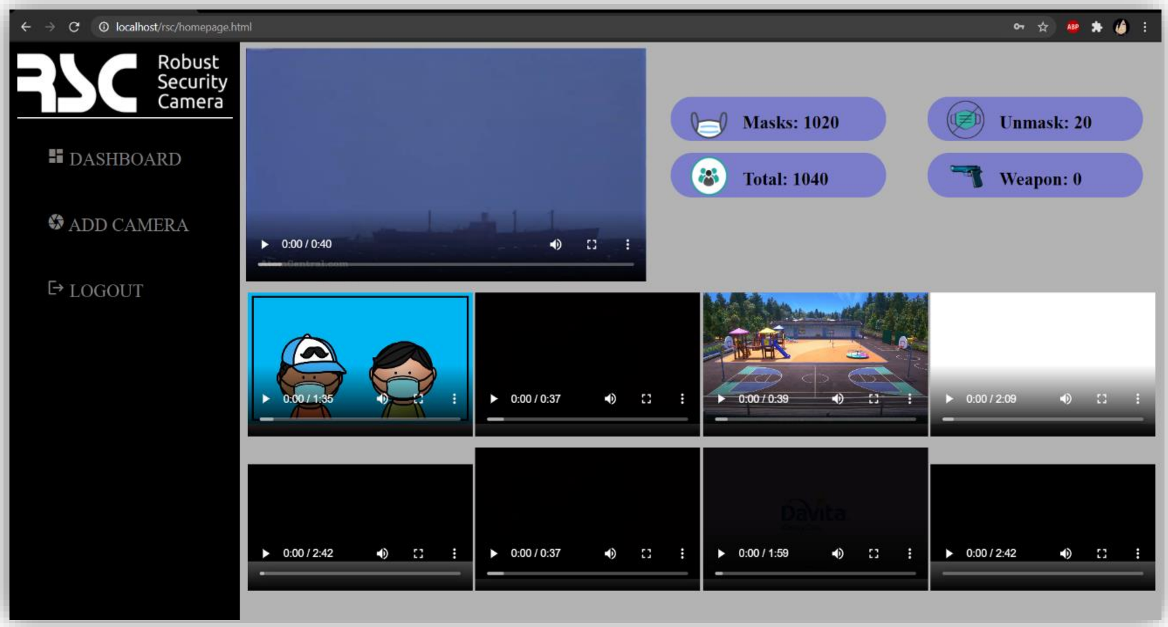


Figure 34 HOME PAGE

5.2 MOBILE INTERFACE

5.2.1 REGISTRATION PAGE

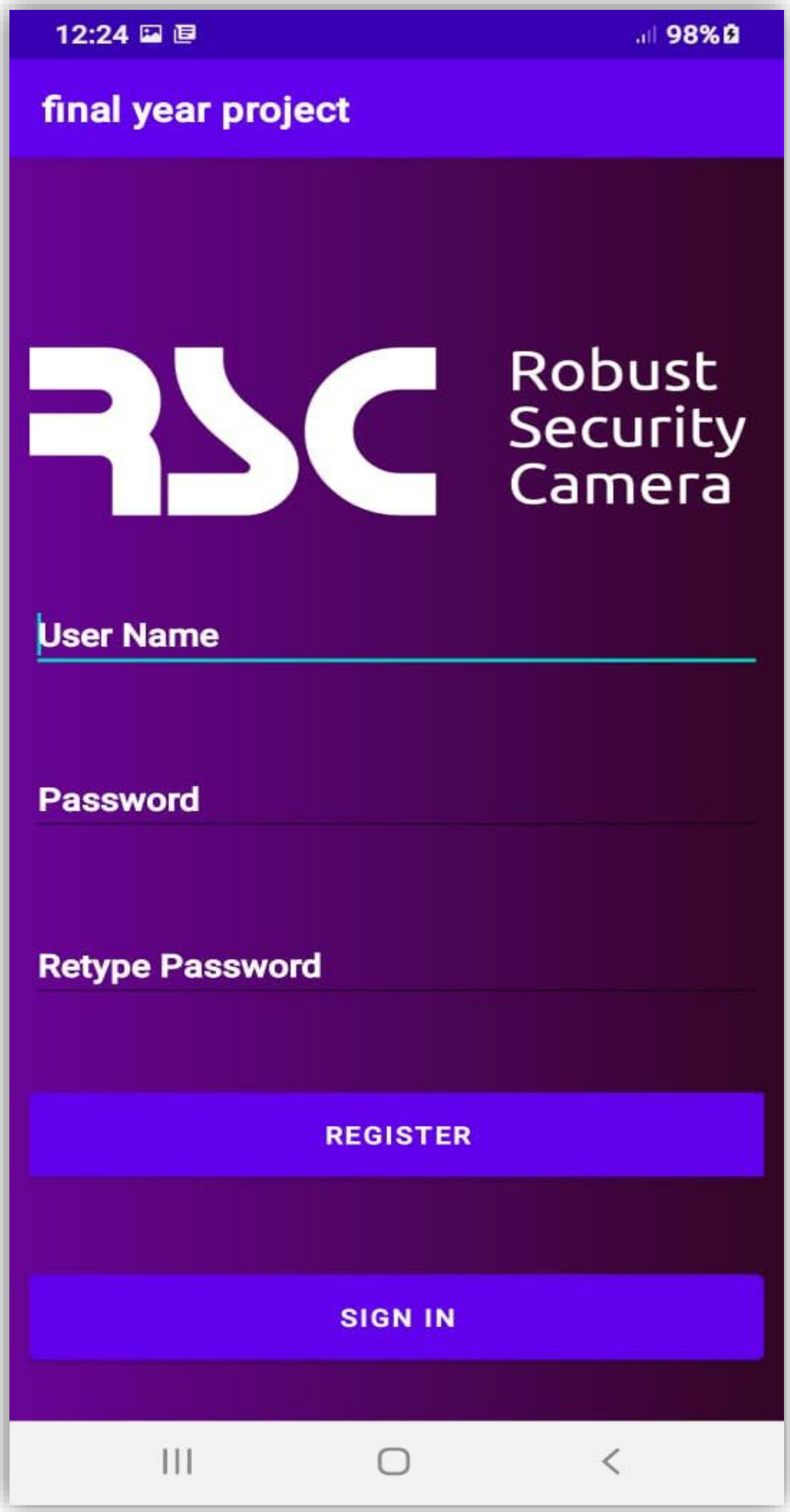


Figure 35 REGISTRATION

5.2.2 LOGIN PAGE

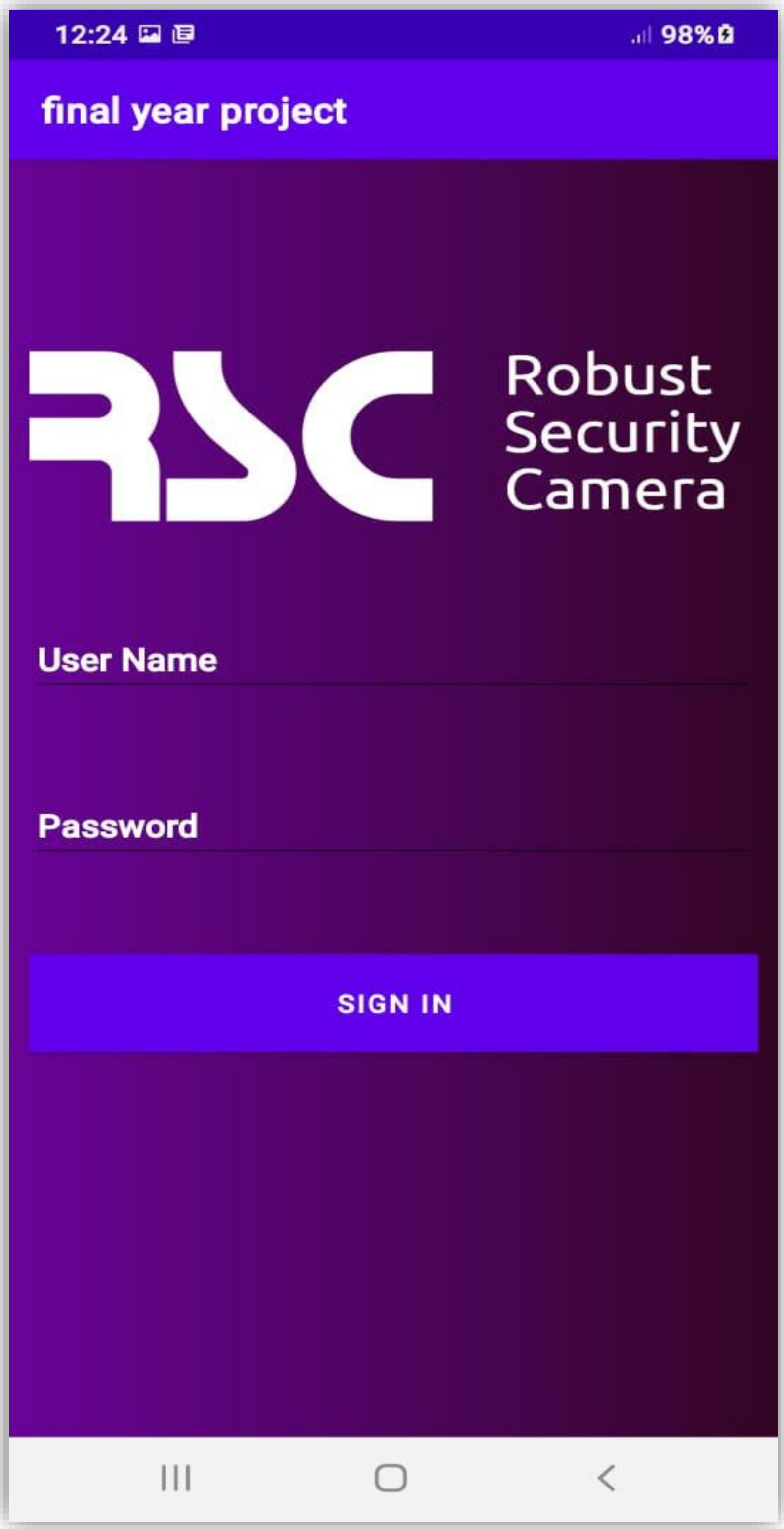
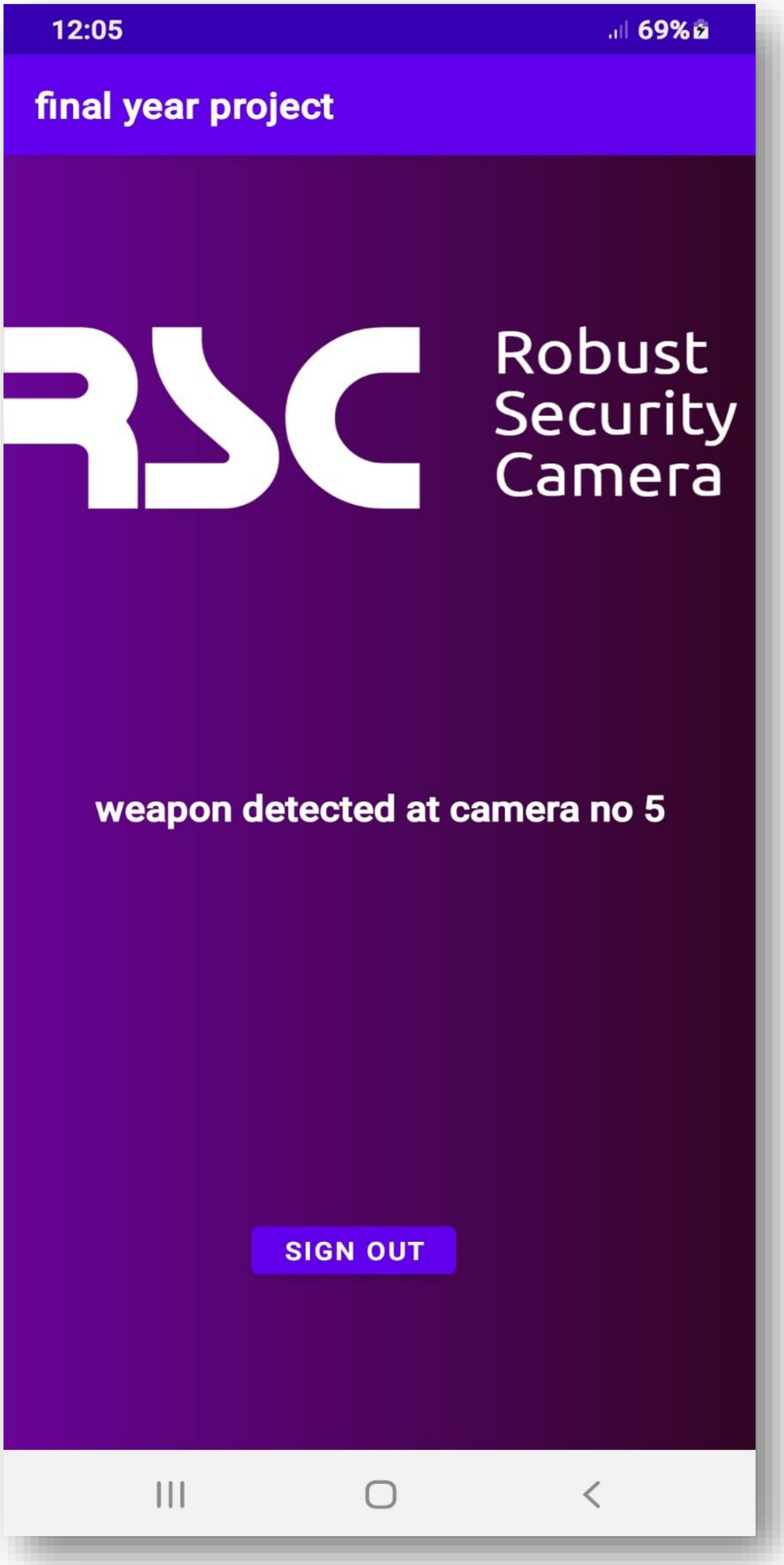


Figure 36 LOGIN

5.2.3 HOME PAGE



12:05

69%

final year project

RSC Robust Security Camera

weapon detected at camera no 5

Figure 37 HOME

CHAPTER: 6 TEST AND RESULTS

6.1 WEAPON DETECTION

6.1.1 TESTING

6.1.1.1 TRAINING & VALIDATION LOSS

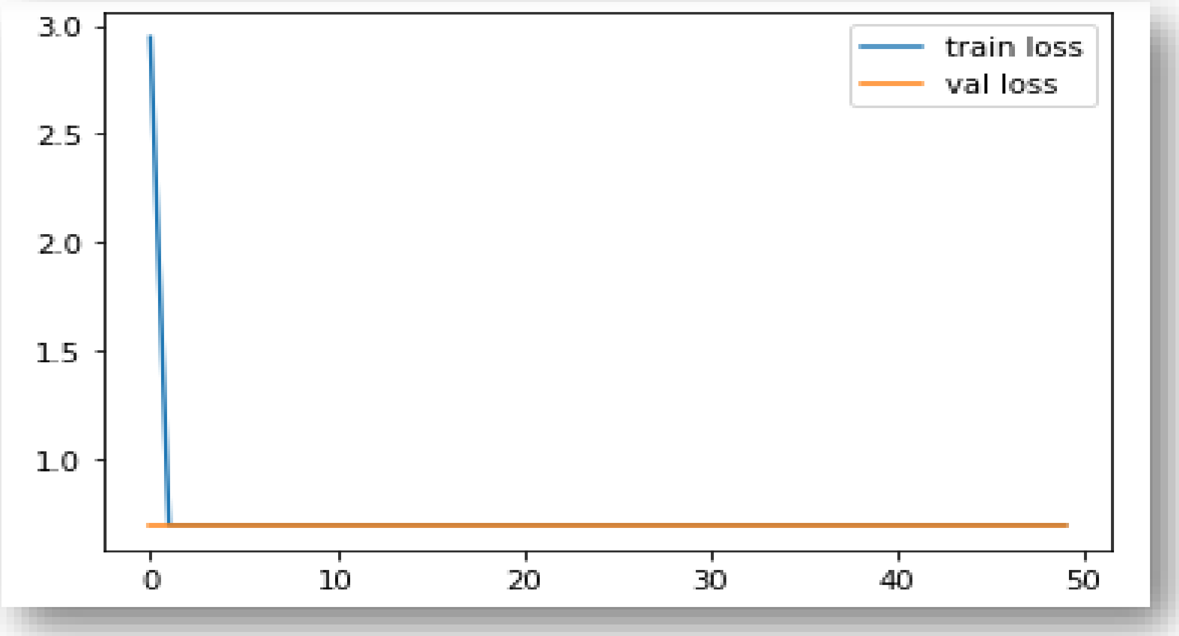


Figure 38 TRAINING & VALIDATION LOSS

6.1.1.2 TRAINING & VALIDATION ACCURACY

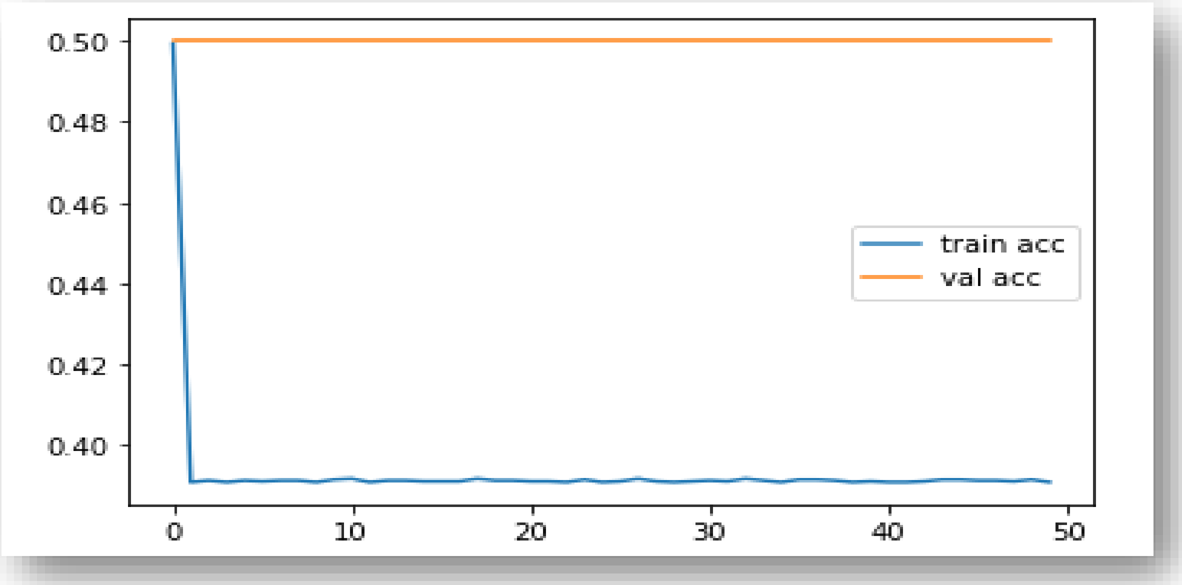


Figure 39 TRAINING & VALIDATION ACCURACY

6.1.2 RESULTS

6.1.2.1 GUN

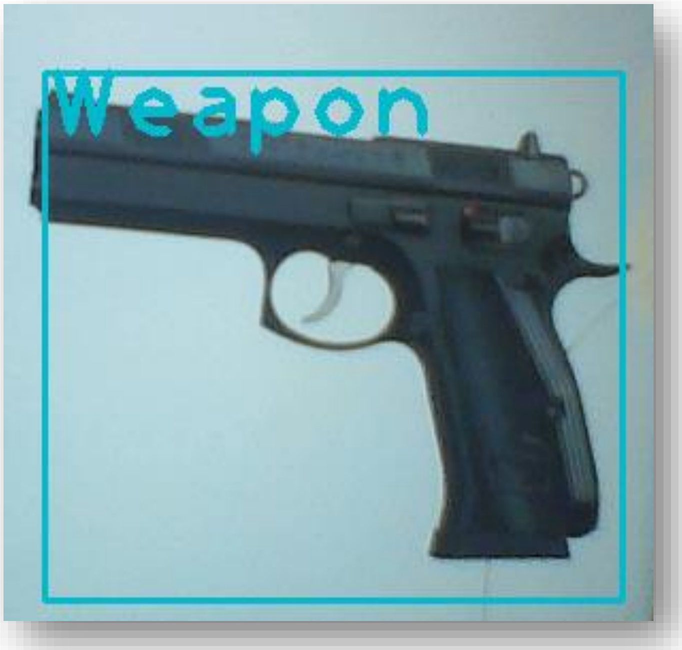


Figure 40 RESULT (PISTOL 1)

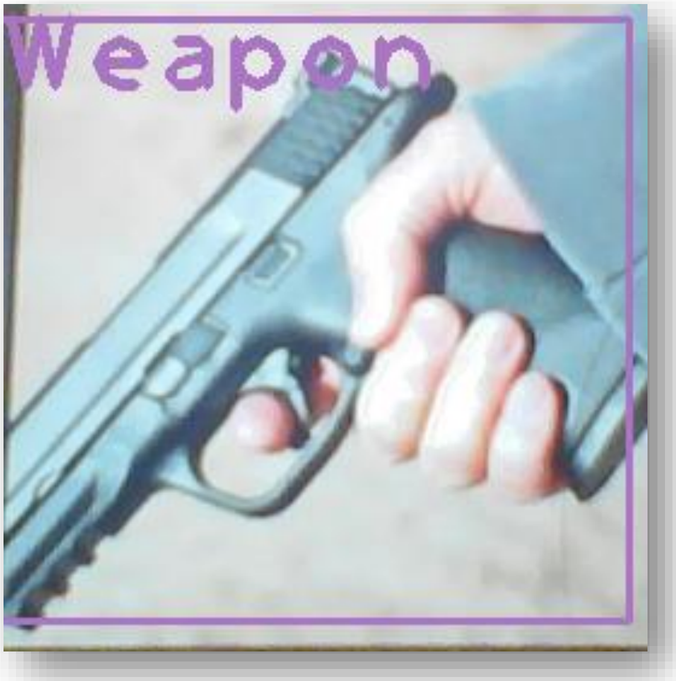


Figure 41 RESULT (PISTOL 2)

6.1.2.2 REVOLVER

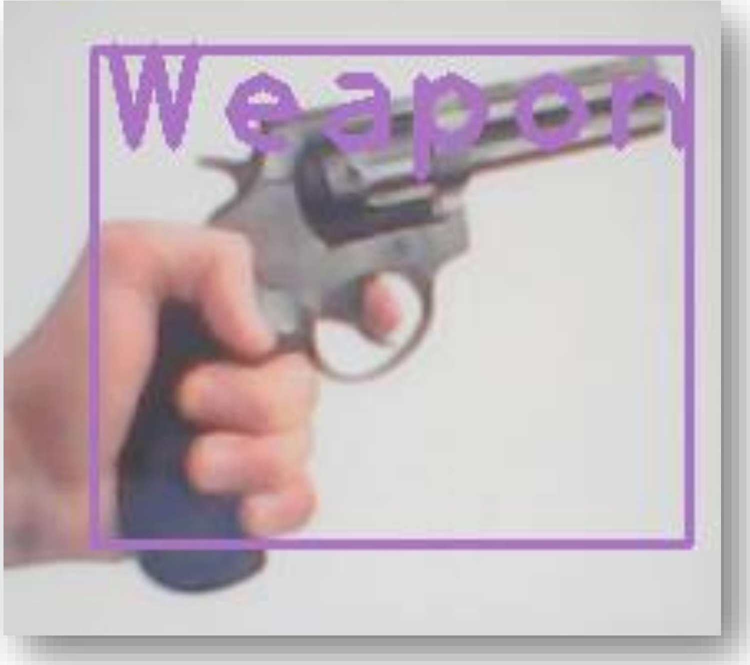


Figure 42 RESULT (REVOLVER)

6.1.2.2 AK47

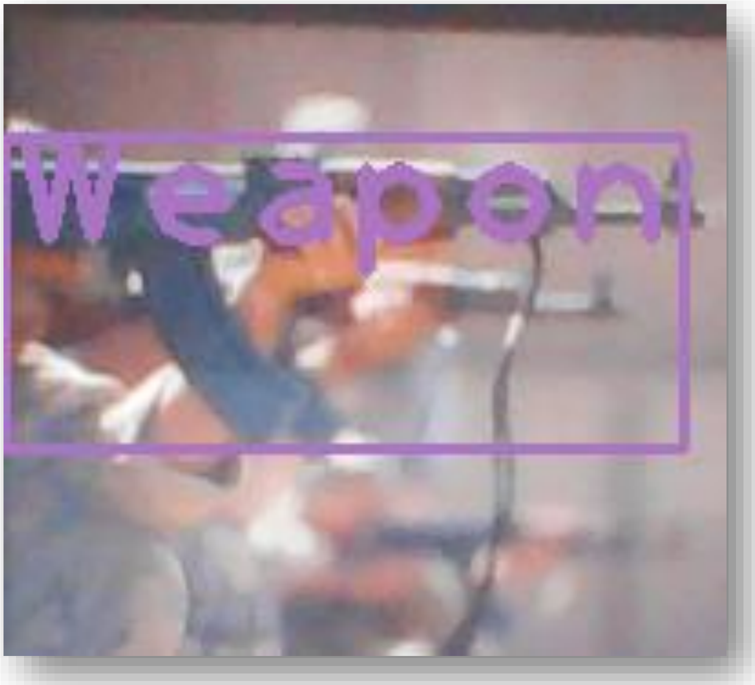


Figure 43 RESULT (AK47)

6.2 FACE MASK DETECTION

6.2.1 TESTING

6.2.1.1 TRAINING & VALIDATION LOSS

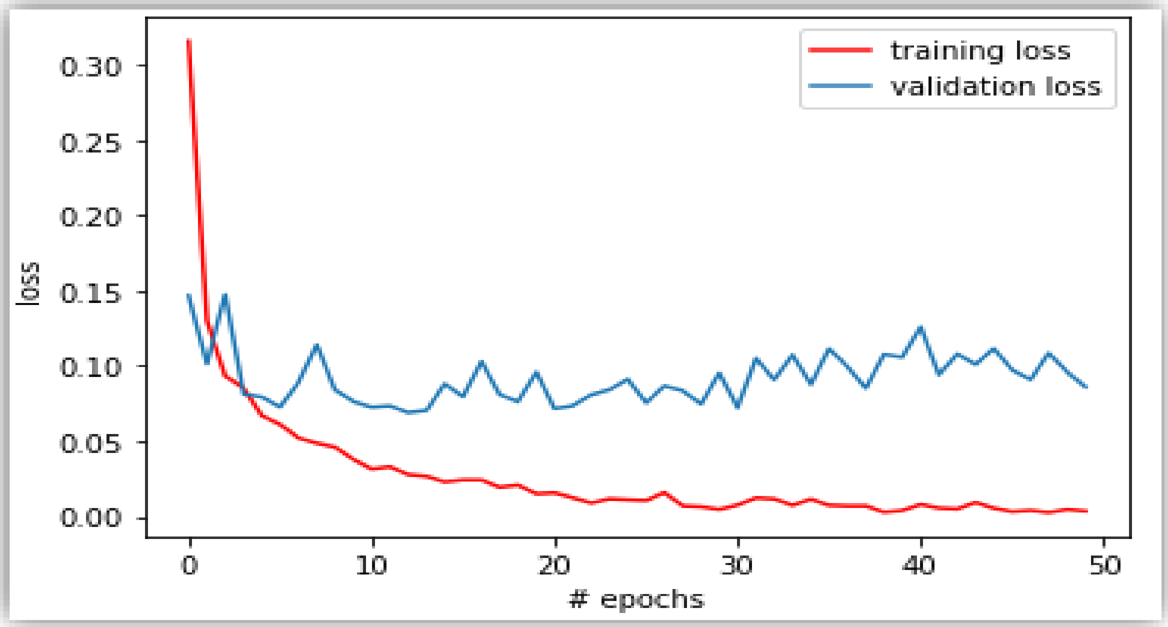


Figure 44 TRAINING & VALIDATION LOSS

6.2.1.2 TRAINING & VALIDATION ACCURACY

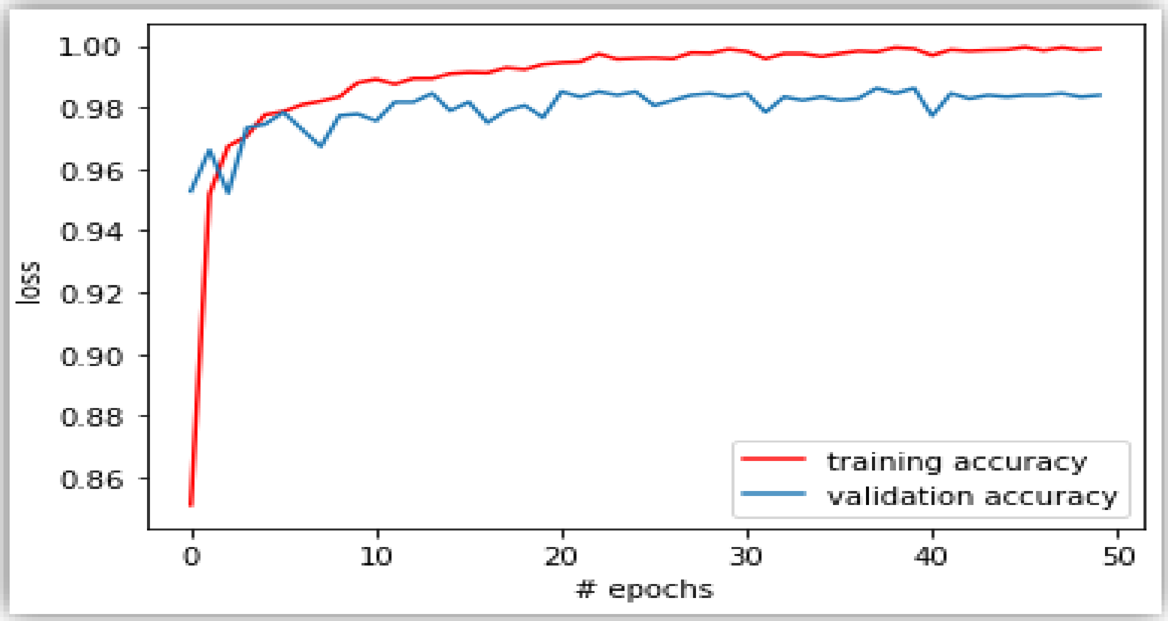


Figure 45 TRAINING & VALIDATION ACCURACY

6.2.2 RESULTS

6.2.2.1 WITH MASK & COUNTER

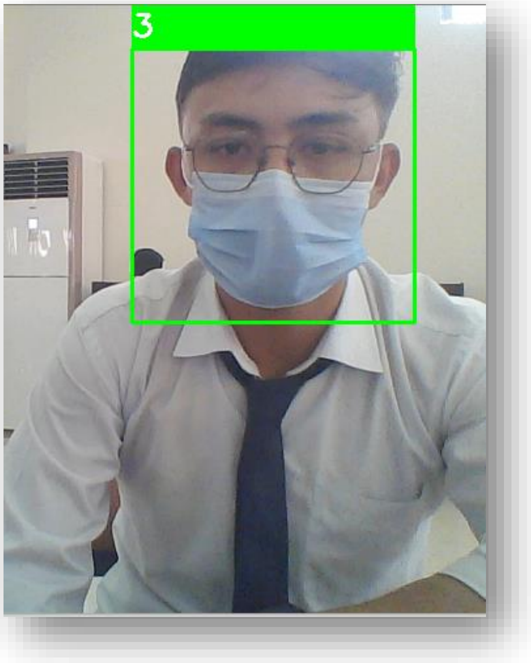


Figure 46 RESULT (WITH MASK & COUNTER)

6.2.2.2 WITHOUT MASK & COUNTER

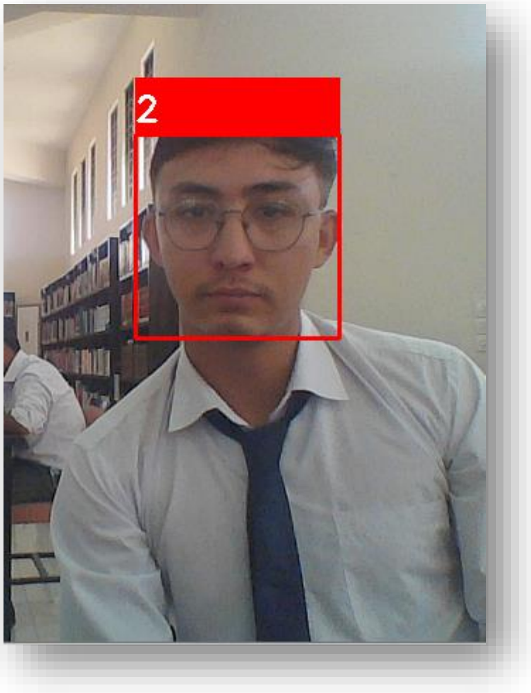


Figure 47 RESULT (WITHOUT MASK & COUNTER)

6.2.2.1 WITH & WITHOUT MASK & COUNTER

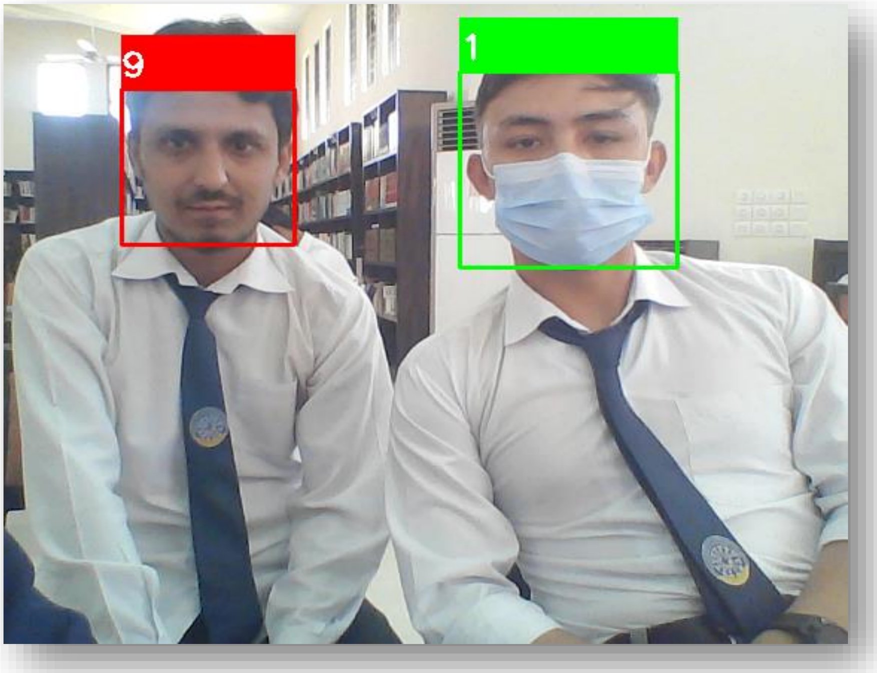


Figure 48 RESULTS (WITH AND WITHOUT MASK & COUNTER)

6.3.1 INTEGRATION

6.3.1.1 RESULTS

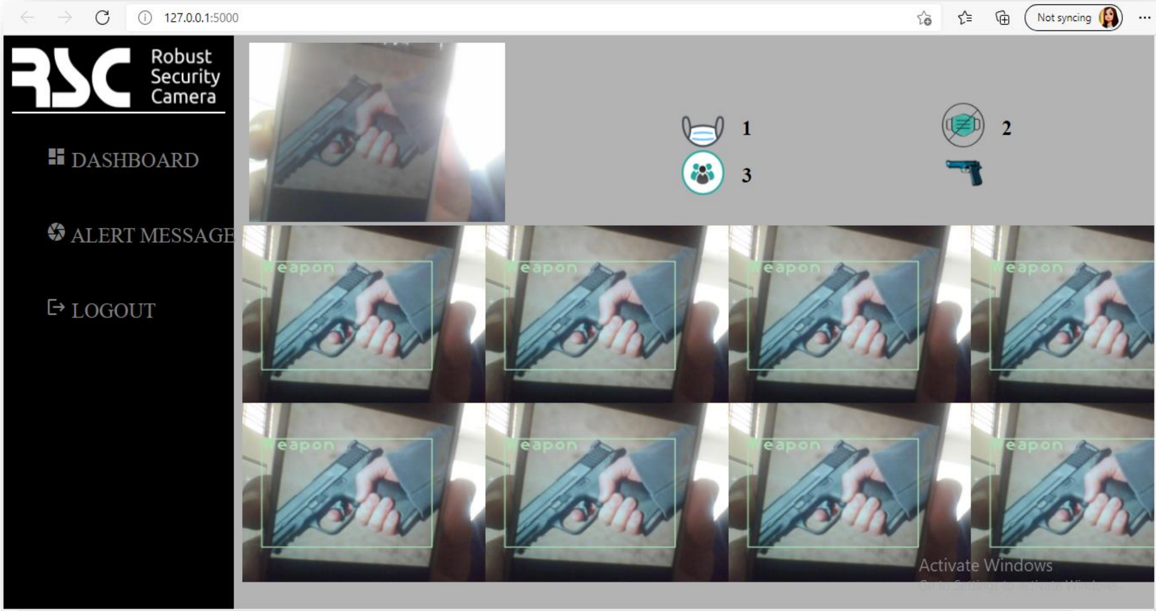


Figure 49 INTEGRATION (WEAPON DETECTED)

CHAPTER: 7 CONCLUSIONS

The purpose of our project is security i.e., detection of weapon and proper application of SOP’S (counting people & entering them along-with face mask detection). So, we used YOLOv3 ML model (with 75% accuracy) for weapon detection and CNN’s Sequential ML model (with 99% accuracy) for Face Mask Detection. We developed two applications i.e. Mobile Application (through android studio) and Web Application (through Visual Studio). We integrated our web application with ML models (Weapon Detection Model & Face Mask Detection Model) using “Flask” as a framework while “Python” as a programming language. And integrated mobile application with web application using “Firebase Cloud Platform” as a framework while “Java” as a programming language.

REFERENCE

[1] M. Mansoor Roomi, “Detection of Concealed Weapons in X-Ray Images Using Fuzzy K-NN,” *Int. J. Comput. Sci. Eng. Inf. Technol.*, vol. 2, no. 2, pp. 187–196, 2012, doi: 10.5121/ijcseit.2012.2216.

[2] M. Grega, A. Matiolański, P. Guzik, and M. Leszczuk, “Automated detection of firearms and knives in a CCTV image,” *Sensors (Switzerland)*, vol. 16, no. 1, 2016, doi: 10.3390/s16010047.

[3] J. Lai and S. Maples, “Developing a Real-Time Gun Detection Classifier,” 2017.

[4] R. Olmos, S. Tabik, and F. Herrera, “Automatic handgun detection alarm in videos using deep learning,” *Neurocomputing*, vol. 275, pp. 66–72, 2018, doi: 10.1016/j.neucom.2017.05.012.

[5] S. Narejo, B. Pandey, D. Esenarro Vargas, C. Rodriguez, and M. R. Anjum, “Weapon Detection Using YOLO V3 for Smart Surveillance System,” *Math. Probl. Eng.*, vol. 2021, 2021, doi: 10.1155/2021/9975700.

[6] A. K. Bhadani and A. Sinha, “A Facemask Detector using Machine Learning and Image Processing Techniques,” *Eng. Sci. Technol. , an Int. J.*, no. November, pp. 0–8, 2020, [Online]. Available:

https://www.researchgate.net/publication/345972030_A_FACEMASK_DETECTOR_USING_MACHINE_LEARNING_AND_IMAGE_PROCESSING_TECHNIQUES.

[7] S. Ge, J. Li, Q. Ye, and Z. Luo, “Detecting masked faces in the wild with LLE-CNNs,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 426–434, 2017, doi: 10.1109/CVPR.2017.53.

[8] M. M. Rahman, M. M. H. Manik, M. M. Islam, S. Mahmud, and J. H. Kim, “An automated system to limit COVID-19 using facial mask detection in smart city network,” *IEMTRONICS 2020 - Int. IOT, Electron. Mechatronics Conf. Proc.*, 2020, doi: 10.1109/IEMTRONICS51293.2020.9216386.

[9] V. Vinitha and V. Velantina, “Covid-19 Facemask Detection With Deep Learning and Computer Vision,” *Int. Res. J. Eng. Technol.*, vol. 07, no. 08, pp. 3127–3132, 2020, [Online]. Available: www.irjet.net.

[10] T. Schlögl, B. Wachmann, W. Kropatsch, and H. Bischof, “Evaluation of People Counting Systems,” *Image Process.*, no. August, 1832.

[11] “People Counting Technology,” *Int. J. Mod. Trends Eng. Res.*, vol. 3, no. 11, pp. 210–214, 2016, doi: 10.21884/ijmter.2016.3142.abprk.

[12] Y. P. Kocak and S. Sevgen, “Detecting and counting people using real-time directional algorithms implemented by compute unified device architecture,” *Neurocomputing*, vol. 248, no. March, pp. 105–111, 2017, doi: 10.1016/j.neucom.2016.08.137.