**Abstract**

In recent years a lot of fake currency note is being printed which have caused great loss and damage towards society. So, it has become a necessity to develop a tool to detect fake currency.

Our proposed system will follow an approach that will detect fake currency note being circulated in our country by using their image. Our project will provide required mobility and compatibility to most peoples as well as credible accuracy for the fake currency detection. We are using image processing to make this application efficient.

This project will find some important features in notes using image processing process which will determine the originality of the currency note.

By using this tool fake notes can easily identify and minimize the count of fake notes in the market.

**Technologies Used**

* Django
* Python
* HTML
* CSS
* JavaScript

**Table Of Contents**

[ABSTRACT](#_TOC_250027) 1

**CHAPTER** **1**

INTRODUCTION……………………………………….5

* 1. [Project Plan](#_TOC_250026)…………………………………………….6
     1. About the project..………………………………6
     2. Purpose and scope………………………………6
  2. [Front End](#_TOC_250023)………………………………………………8
     1. Introduction Of HTML………………………….8
     2. Introduction of CSS…………………………….12
     3. Introduction of JavaScript……………………...15
  3. Back End……………………………………………...18
     1. Python………………………………………….18
     2. Django………………………………………….22

CHAPETR 2

**SOFTWARE DEVELOPMENT LIFE CYCLE………25**

* 1. [Requirement Analysis Phase…......................................26](#_TOC_250016)
     1. [System Requirement Specification 2](#_TOC_250015)7
     2. [Hardware and Software Requirement](#_TOC_250014) 28
     3. [Functional Requirement](#_TOC_250013) 28
     4. [Non-Functional Requirement](#_TOC_250012) 29
     5. [Feasibility Study](#_TOC_250011) 31
     6. [Technical Feasibility](#_TOC_250010) 32
     7. Economic Feasibility 32
     8. Behavioural Feasibility 33

CHAPETR 3

[**DIAGRAM**](#_TOC_250017)**S**……………………………………….….34

1. [Data Flow Diagram](#_TOC_250008)………………………………...35
2. [Sequence Diagram](#_TOC_250007)……………………………........37
3. Component Diagram………………………………38
4. Use Case Diagram……………………………........39

CHAPTER 4

**CODING**……………………………………………..40

* 1. PYTHON Code…………………………..…......41
  2. HTML Code…………………………………….44

CHAPTER 5

**TESTING**………………………………………….....47

5.1 Unit Testing……………………………………...48

* 1. Integration Testing……………………………....49
  2. System Testing…………………………………,.50

5.4 Functional Testing……………………………....51

CHAPETR 6

[SNAPSHOTS](#_TOC_250000)……………………………………….52

CHAPETR 7

[FUTURE](#_TOC_250000) ENHANCEMENT……………………....56

CONCLUSION……………………..........................57

[**REFERENCES**](#_TOC_250000)**…**…………………………………………………………………………..….…….58

**CHAPTER 1**

INTRODUCTION

**Introduction**

* 1. **PROJECT PLAN**

## ABOUT THE PROJECT

Fake currency detection is a serious issue worldwide, affecting the economy of almost every country including India. Currency duplication also known as counterfeit currency is a vulnerable threat on economy. It is now a common phenomenon due to advanced printing and scanning technology. The possible solutions are to use either chemical properties of the currency or to use its physical appearance.

The approach used in our proposed system is based upon physical appearance of the currency. Image processing algorithms have been adopted to extract the features such as kurtosis, skew and entropy which will identify the pattern in the currency note which will determine the originality of the note. Hence, we propose a more user friendly and portable solution to this problem in form of an web app.

## PURPOSE AND SCOPE

The primary purpose of the fake currency detection system is to eliminate the fake notes from the market. This will help the government to clear the fake notes.

There are different other ways to identify the fake note but computer application these days are performing well in almost all fields. This will help in identify fake notes in easiest way.

The main reason to use computer application over the traditional way is –

This application can be used by government in the future also it can be used by merchants to identify the fake notes.

## Front End

## 1.2.1 Introduction of HTML

HTML is an acronym which stands for **Hyper Text Markup Language** which is used for creating web pages and web applications. Let's see what is meant by Hypertext Markup Language, and Web page.

**Hyper Text:**

HyperText simply means "Text within Text." A text has a link within it, is a hypertext. Whenever you click on a link which brings you to a new webpage, you have clicked on a hypertext. HyperText is a way to link two or more web pages (HTML documents) with each other.

**Markup language:**

A markup language is a computer language that is used to apply layout and formatting conventions to a text document. Markup language makes text more interactive and dynamic. It can turn text into images, tables, links, etc.

**Web Page:**

A web page is a document which is commonly written in HTML and translated by a web browser. A web page can be identified by entering an URL. A Web page can be of the static or dynamic type. **With the help of HTML only, we can create static web pages**.

Hence, HTML is a markup language which is used for creating attractive web pages with the help of styling, and which looks in a nice format on a web browser. An HTML document is made of many HTML tags and each HTML tag contains different content.

**Brief History of HTML:**

In the late 1980's , a physicist, Tim Berners-Lee who was a contractor at CERN, proposed a system for CERN researchers. In 1989, he wrote a memo proposing an internet based hypertext system.

Tim Berners-Lee is known as the father of HTML. The first available description of HTML was a document called "HTML Tags" proposed by Tim in late 1991. The latest version of HTML is HTML5, which we will learn later in this tutorial.

**HTML Versions:**

Since the time HTML was invented there are lots of HTML versions in market, the brief introduction about the HTML version is given below:

**HTML 1.0:** The first version of HTML was 1.0, which was the barebones version of HTML language, and it was released in1991.

**HTML 2.0:** This was the next version which was released in 1995, and it was standard language version for website design. HTML 2.0 was able to support extra features such as form-based file upload, form elements such as text box, option button, etc.

**HTML 3.2:** HTML 3.2 version was published by W3C in early 1997. This version was capable of creating tables and providing support for extra options for form elements. It can also support a web page with complex mathematical equations. It became an official standard for any browser till January 1997. Today it is practically supported by most of the browsers.

**HTML 4.01:** HTML 4.01 version was released on December 1999, and it is a very stable version of HTML language. This version is the current official standard, and it provides added support for stylesheets (CSS) and scripting ability for various multimedia elements.

**HTML5 :** HTML5 is the newest version of HyperText Markup language. The first draft of this version was announced in January 2008. There are two major organizations one is W3C (World Wide Web Consortium), and another one is WHATWG( Web Hypertext Application Technology Working Group) which are involved in the development of HTML 5 version, and still, it is under development.

**Features of HTML**

1) It is a very **easy and simple language**. It can be easily understood and modified.

2) It is very easy to make an **effective presentation** with HTML because it has a lot of formatting tags.

3) It is a **markup language**, so it provides a flexible way to design web pages along with the text.

4) It facilitates programmers to add a **link** on the web pages (by html anchor tag), so it enhances the interest of browsing of the user.

5) It is **platform-independent** because it can be displayed on any platform like Windows, Linux, and Macintosh, etc.

6) It facilitates the programmer to add **Graphics, Videos, and Sound** to the web pages which makes it more attractive and interactive.

7) HTML is a case-insensitive language, which means we can use tags either in lower-case or upper-case.

## Introduction of CSS

CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. It can also be used with any kind of XML documents including plain XML, SVG and XUL.

CSS is used along with HTML and JavaScript in most websites to create user interfaces for web applications and user interfaces for many mobile applications.

* **What does CSS do**

You can add new looks to your old HTML documents.

You can completely change the look of your website with only a few changes in CSS code.

* **Why use CSS**

These are the three major benefits of CSS:

**1) Solves a big problem**

Before CSS, tags like font, color, background style, element alignments, border and size had to be repeated on every web page. This was a very long process. For example: If you are developing a large website where fonts and color information are added on every single page, it will be become a long and expensive process. CSS was created to solve this problem. It was a W3C recommendation.

**2) Saves a lot of time**

CSS style definitions are saved in external CSS files so it is possible to change the entire website by changing just one file.

**3) Provide more attributes**

CSS provides more detailed attributes than plain HTML to define the look and feel of the website.

* **Advantages of CSS**

**CSS saves time** − You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.

**Pages load faster** − If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.

**Easy maintenance** − To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.

**Superior styles to HTML** − CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.

**Multiple Device Compatibility** − Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.

**Global web standards** − Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

* **CSS Versions**

Cascading Style Sheets level 1 (CSS1) came out of W3C as a recommendation in December 1996. This version describes the CSS language as well as a simple visual formatting model for all the HTML tags.

CSS2 became a W3C recommendation in May 1998 and builds on CSS1. This version adds support for media-specific style sheets e.g. printers and aural devices, downloadable fonts, element positioning and tables.

**1.2.3 Introduction of JavaScript**

JavaScript is a light-weight object-oriented programming language which is used by several websites for scripting the webpages. It is an interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document. It was introduced in the year 1995 for adding programs to the webpages in the Netscape Navigator browser. Since then, it has been adopted by all other graphical web browsers. With JavaScript, users can build modern web applications to interact directly without reloading the page every time. The traditional website uses js to provide several forms of interactivity and simplicity.

Although, JavaScript has no connectivity with Java programming language. The name was suggested and provided in the times when Java was gaining popularity in the market. In addition to web browsers, databases such as CouchDB and MongoDB uses JavaScript as their scripting and query language.

* **Features of JavaScript**

There are following features of JavaScript:

* All popular web browsers support JavaScript as they provide built-in execution environments.
* JavaScript follows the syntax and structure of the C programming language. Thus, it is a structured programming language.
* JavaScript is a weakly typed language, where certain types are implicitly cast (depending on the operation).
* JavaScript is an object-oriented programming language that uses prototypes rather than using classes for inheritance.
* It is a light-weighted and interpreted language.
* It is a case-sensitive language.
* JavaScript is supportable in several operating systems including, Windows, macOS, etc.
* It provides good control to the users over the web browsers.
* **History of JavaScript**

In 1993, **Mosaic**, the first popular web browser, came into existence. In the **year 1994**, **Netscape** was founded by **Marc Andreessen**. He realized that the web needed to become more dynamic. Thus, a 'glue language' was believed to be provided to HTML to make web designing easy for designers and part-time programmers. Consequently, in 1995, the company recruited **Brendan Eich** intending to implement and embed Scheme programming language to the browser. But, before Brendan could start, the company merged with **Sun Microsystems** for adding Java into its Navigator so that it could compete with Microsoft over the web technologies and platforms. Now, two languages were there: Java and the scripting language. Further, Netscape decided to give a similar name to the scripting language as Java's. It led to 'Javascript'. Finally, in May 1995, Marc Andreessen coined the first code of Javascript named '**Mocha**'. Later, the marketing team replaced the name with '**LiveScript**'. But, due to trademark reasons and certain other reasons, in December 1995, the language was finally renamed to 'JavaScript'. From then, JavaScript came into existence.

* **Application of JavaScript**
* JavaScript is used to create interactive websites. It is mainly used for:
* Client-side validation,
* Dynamic drop-down menus,
* Displaying date and time,
* Displaying pop-up windows and dialog boxes (like an alert dialog box, confirm dialog box and prompt dialog box),
* Displaying clocks etc.

## Back End

## 1.3.1 Introduction of Python

Python tutorial provides basic and advanced concepts of Python. Our Python tutorial is designed for beginners and professionals.

Python is a simple, general purpose, high level, and object-oriented programming language.

Python is an interpreted scripting language also. Guido Van Rossum is known as the founder of Python programming.

* **What is Python**

**Python** is a general purpose, dynamic, [high-level](https://www.javatpoint.com/classification-of-programming-languages), and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures.

Python is easy to learn yet powerful and versatile scripting language, which makes it attractive for Application Development.

Python's syntax and dynamic typing with its interpreted nature make it an ideal language for scripting and rapid application development.

Python supports multiple programming pattern, including object-oriented, imperative, and functional or procedural programming styles.

Python is not intended to work in a particular area, such as web programming. That is why it is known as multipurpose programming language because it can be used with web, enterprise, 3D CAD, etc.

We don't need to use data types to declare variable because it is dynamically typed so we can write a=10 to assign an integer value in an integer variable.

Python makes the development and debugging fast because there is no compilation step included in Python development, and edit-test-debug cycle is very fast.

* **Python 2 vs. Python 3**

In most of the programming languages, whenever a new version releases, it supports the features and syntax of the existing version of the language, therefore, it is easier for the projects to switch in the newer version. However, in the case of Python, the two versions Python 2 and Python 3 are very much different from each other.

A list of differences between Python 2 and Python 3 are given below:

Python 2 uses **print** as a statement and used as print "something" to print some string on the console. On the other hand, Python 3 uses **print** as a function and used as print("something") to print something on the console.

Python 2 uses the function raw\_input() to accept the user's input. It returns the string representing the value, which is typed by the user. To convert it into the integer, we need to use the int() function in Python. On the other hand, Python 3 uses input() function which automatically interpreted the type of input entered by the user. However, we can cast this value to any type by using primitive functions (int(), str(), etc.).

In Python 2, the implicit string type is ASCII, whereas, in Python 3, the implicit string type is Unicode.

Python 3 doesn't contain the xrange() function of Python 2. The xrange() is the variant of range() function which returns a xrange object that works similar to Java iterator. The range() returns a list for example the function range(0,3) contains 0, 1, 2.

There is also a small change made in Exception handling in Python 3. It defines a keyword **as** which is necessary to be used. We will discuss it in Exception handling section of Python programming tutorial.

* **Python History**

Python was invented by **Guido van Rossum** in 1991 at CWI in Netherland. The idea of Python programming language has taken from the ABC programming language or we can say that ABC is a predecessor of Python language.

There is also a fact behind the choosing name Python. Guido van Rossum was a fan of the popular BBC comedy show of that time, **"Monty Python's Flying Circus"**. So he decided to pick the name **Python** for his newly created programming language.

Python has the vast community across the world and releases its version within the short period.

* **Where is Python used?**

Python is a general-purpose, popular programming language and it is used in almost every technical field. The various areas of Python use are given below.

* Data Science
* Date Mining
* Desktop Applications
* Console-based Applications
* Mobile Applications
* Software Development
* Artificial Intelligence
* Web Applications
* Enterprise Applications
* 3D CAD Applications
* Machine Learning

**1.3.2 Django**

Django is a web application framework written in Python programming language. It is based on MVT (Model View Template) design pattern. The Django is very demanding due to its rapid development feature. It takes less time to build application after collecting client requirement.

This framework uses a famous tag line:

**The web framework for perfectionists with deadlines.**

By using Django, we can build web applications in very less time. Django is designed in such a manner that it handles much of configure things automatically, so we can focus on application development only.

* **History**

Django was design and developed by Lawrence journal world in 2003 and publicly released under BSD license in July 2005. Currently, DSF (Django Software Foundation) maintains its development and release cycle.

Django was released on 21, July 2005. Its current stable version is 2.0.3 which was released on 6 March, 2018.

* **Popularity**

Django is widely accepted and used by various well-known sites such as:

* Instagram
* Mozilla
* Disqus
* Pinterest
* Bitbucket
* The Washington Times
* **Features of Django**
* Rapid Development
* Secure
* Scalable
* Fully loaded
* Versatile
* Open Source
* Vast and Supported Community
* **Rapid Development**

Django was designed with the intention to make a framework which takes less time to build web application. The project implementation phase is a very time taken but Django creates it rapidly.

* **Secure**

Django takes security seriously and helps developers to avoid many common security mistakes, such as SQL injection, cross-site scripting, cross-site request forgery etc. Its user authentication system provides a secure way to manage user accounts and passwords.

* **Scalable**

Django is scalable in nature and has ability to quickly and flexibly switch from small to large scale application project.

* **Fully loaded**

Django includes various helping task modules and libraries which can be used to handle common Web development tasks. Django takes care of user authentication, content administration, site maps, RSS feeds etc.

* **Versatile**

Django is versatile in nature which allows it to build applications for different-different domains. Now a days, Companies are using Django to build various types of applications like: content management systems, social networks sites or scientific computing platforms etc.

* **Open Source**

Django is an open source web application framework. It is publicly available without cost. It can be downloaded with source code from the public repository. Open source reduces the total cost of the application development.

**CHAPTER 2**

**SOFTWARE DEVELOPMENT LIFE CYCLE**

## REQUIREMENT ANALYSIS PHASE

The Requirements Analysis Phase begins when the previous phase objectives have been achieved. Documentation related to user requirements from the Concept Development Phase and the Planning Phase shall be used as the basis for further user needs analysis and the development of detailed requirements. Multiple-release projects require only one iteration of the Requirements Analysis Phase, which should involve requirements definition for all planned releases.

The objective of this phase is to define in more detail the system inputs, processes, outputs and interfaces. At the end of this phase the system’s processes will be defined at the functional level, meaning the functions to be performed will be known, but not necessarily how they will be performed. Unless specifically constrained by the Project Charter, Requirements Analysis should not consider the computer programs, files and data streams. Requirements Analysis will identify and consider the risks related to how the technology will be integrated into the standard operating procedures. Requirements Analysis will collect the functional and system requirements of the business process, the user requirements and the operational requirements (e.g., when operational what is necessary to keep the system up and running).

## System Requirement Specification

Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers on how the software product should function (in a market-driven project, these roles may be played by the marketing and development divisions). Software requirements specification is a rigorous assessment of requirements before the more specific system design stages, and its goal is to reduce later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules. Used appropriately, software requirements specifications can help prevent software project failure.

The software requirements specification document lists sufficient and necessary requirements for the project development. To derive the requirements, the developer needs to have clear and thorough understanding of the products under development. This is achieved through detailed and continuous communications with the project team and customer throughout the software development process.

## Purpose

The purpose of this document is to give a detailed description of the requirements for the “Amazing Lunch Indicator” (ALI) software. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.

## Hardware and Software Requirement

#### Hardware Specification

* + - * RAM 4 GB
      * Windows 10

#### Software Requirements:

* + - * Python
      * PyCharm
      * Browser to Test

## Functional Requirement

In Software engineering and systems engineering, a functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behaviour, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define whata system is supposed to accomplish. Behavioural requirements describing all the cases where the system uses the functional requirements are captured in use cases. Functional requirements are supported by non-functional requirements which impose constraints on the design or implementation.

As defined in requirements engineering, functional requirements specify particular results of a system. This should be contrasted with non-functional requirements which specify overall characteristics such as cost and reliability. Functional requirements drive the application architecture of a system, while non-functional requirements drive the technical architecture of a system.

## Non-Functional Requirement

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours.

They are contrasted with functional requirements that define specific behaviour or functions. The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture, because they are usually Architecturally Significant Requirements.

Broadly, functional requirements define what a system is supposed to do and non-functional requirements define how a system is supposed to be. Functional requirements are usually in the form of, an individual action or part of the system, perhaps explicitly in the sense of a mathematical function, a black box description input, output, process and control functional model or IPO Model. In contrast, non-functional requirements are in the form of, an overall property of the system as a whole or of a particular aspect and not a specific function. The system's overall properties commonly mark the difference between whether the development project has succeeded or failed.

Non-functional requirements are often called "quality attributes" of a system. Other terms for non-functional requirements are "qualities", "quality goals", "quality of service requirements", "constraints" and "non-behavioural requirements". Informally these are sometimes called the "ileitis", from attributes like stability and portability. Qualities—that is non-functional requirements—can be divided into two main categories:

1. Execution qualities, such as safety, security and usability, which are observable during operation.
2. Evolution qualities, such as testability, maintainability, extensibility and scalability, which are embodied in the static structure of the system.

## Feasibility Study

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness.

A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus when a new application is proposed it normally goes through a feasibility study before it is approved for development. Following are its feature

**2.1.6 TECHNICAL FEASIBILITY**

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures.

Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

* Does the existing technology sufficient for the suggested one?
* Can the system expand if developed?

#### ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest.

One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

* The costs conduct a full system investigation.
* The cost of the hardware and software.
* The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development

#### BEHAVIORAL FEASIBILITY

This includes the following questions:

* Is there sufficient support for the users?
* Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioural aspects are considered carefully and conclude that the project is behaviourally feasible.

**CHAPTER 3**

DIAGRAMS

**Implementation or Architecture Diagrams**

1. **Data Flow Diagram (DFD)**

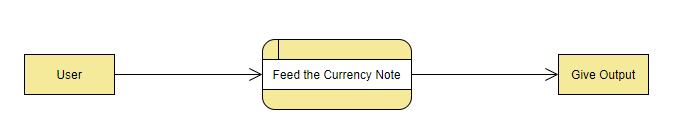
A data flow diagram is a graphical representation that depicts information flow and the transforms that are applied as data move from input to output. The basic form of a data flow diagram, also known as a data flow graph or a bubble chart, The data flow diagram may be used to represent a system or software at any level of abstraction. As information moves through software, it is modified by a series of transformations.

A data flow diagram is a graphical representation that depicts information flow and the transforms that are applied as data move from input to output. The basic form of a data flow diagram, also known as a data flow graph or a bubble chart. DFD is an abstract description of the system. The data flow diagram may be used to represent a system or software at any level of abstraction. DFDs may be partitioned into levels that represent increasing information flow and functional detail. Therefore, the DFD provides a mechanism for functional modeling as well as information flow modeling.

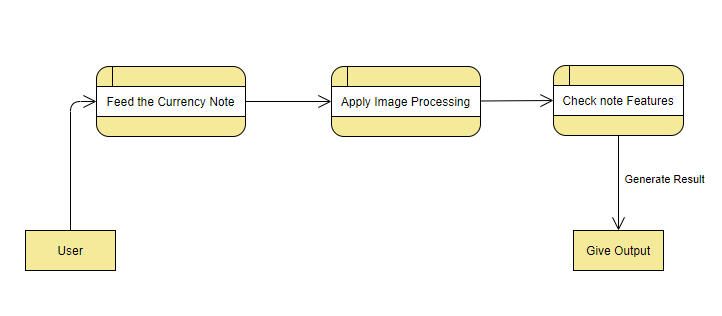
DFDs are very useful in understanding a system and can be effectively used during analysis. DFDs can be hierarchically organized, which helps in progressively partitioning and analyzing large systems. Such DFDs are called leveled DFDs.

.

**Level-0 DFD**

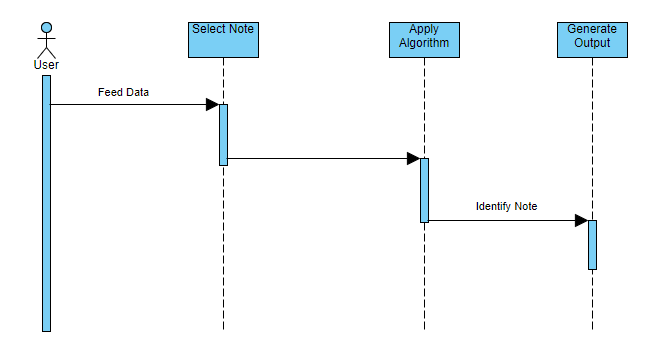
****

**Level-1 DFD**

****

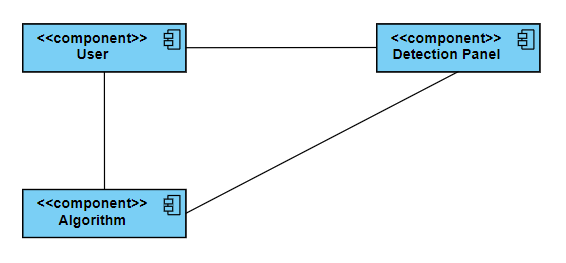
**2. Sequence Diagram**

A **sequence diagram** simply depicts interaction between objects in a **sequential** order i.e. the order in which these interactions take place. We can also use the terms event **diagrams** or event scenarios to refer to a **sequence diagram**. **Sequence diagrams** describe how and in what order the objects in a system function.



**3. Component Diagram**

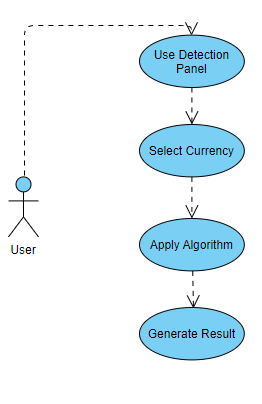
A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development. A component diagram allows verification that a system's required functionality is acceptable. These diagrams are also used as a communication tool between the developer and stakeholders of the system. Programmers and developers use the diagrams to formalize a roadmap for the implementation, allowing for better decision-making about task assignment or needed skill improvements.

****

**4. Use Case Diagram**

To model a system, the most important aspect is to capture the dynamic behavior. Dynamic behavior means the behavior of the system when it is running/operating.

Only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior. In UML, there are five diagrams available to model the dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature, there should be some internal or external factors for making the interaction. These internal and external agents are known as actors. Use case diagrams consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system. Hence to model the entire system, a number of use case diagrams are used.



**CHAPTER 4**

CODING

1. **Python Code**

from django.contrib.auth import authenticate, login

from django.shortcuts import render, redirect

import pickle

from django.conf import settings

import os

from pathlib import Path

import cv2

import numpy as np

import pathlib

from PIL import Image

import os, os.path

import glob

import base64

from io import StringIO

import io

import cv2

import base64

import numpy as np

from PIL import Image

import matplotlib.pyplot as plt

import pyautogui as pag

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import confusion\_matrix

import cv2

from scipy.stats import kurtosis, skew,entropy

import numpy as np

from scipy import ndimage

import statistics

import base64

BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent

def index(request):

if request.method == "POST":

data = pd.read\_csv('banknote\_authentication.txt', header=None)

data.columns = ['var', 'skew', 'curt', 'entr', 'auth']

print(data.head())

sns.pairplot(data, hue='auth')

sns.countplot(x=data['auth'])

target\_count = data.auth.value\_counts()

nb\_to\_delete = target\_count[0] - target\_count[1]

data = data.sample(frac=1, random\_state=42).sort\_values(by='auth')

data = data[nb\_to\_delete:]

x = data.loc[:, data.columns != 'auth']

y = data.loc[:, data.columns == 'auth']

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.3, random\_state=42)

scalar = StandardScaler()

scalar.fit(x\_train)

x\_train = scalar.transform(x\_train)

x\_test = scalar.transform(x\_test)

clf = LogisticRegression(solver='lbfgs', random\_state=42, multi\_class='auto')

clf.fit(x\_train, y\_train.values.ravel())

y\_pred = np.array(clf.predict(x\_test))

conf\_mat = pd.DataFrame(confusion\_matrix(y\_test, y\_pred),

columns=["Pred.Negative", "Pred.Positive"],

index=['Act.Negative', "Act.Positive"])

tn, fp, fn, tp = confusion\_matrix(y\_test, y\_pred).ravel()

accuracy = round((tn+tp)/(tn+fp+fn+tp), 4)

try:

my\_uploaded\_file = request.FILES['my\_uploaded\_file'].read()

my\_uploaded\_file\_base64 = base64.b64encode(my\_uploaded\_file)

print('loaded')

def stringToImage(base64\_string):

imgdata = base64.b64decode(base64\_string)

image = Image.open(io.BytesIO(imgdata))

return image

def stringToEdgeImage(base64\_string):

imgdata = base64.b64decode(base64\_string)

image = Image.open(io.BytesIO(imgdata))

#img\_gray = cv2.cvtColor(np.array(image), cv2.COLOR\_BGR2GRAY)

img\_blur = cv2.GaussianBlur(np.array(image), (3,3), 0)

sobelxy = cv2.Sobel(src=img\_blur, ddepth=cv2.CV\_64F, dx=1, dy=0, ksize=5)

return np.array(sobelxy)

######

opencvImage = cv2.cvtColor(np.array(stringToImage(my\_uploaded\_file\_base64)), cv2.COLOR\_RGB2BGR)

norm\_image = cv2.normalize(opencvImage, None, alpha=0, beta=1, norm\_type=cv2.NORM\_MINMAX, dtype=cv2.CV\_32F)

img\_blur = cv2.GaussianBlur(norm\_image, (3,3), 0)

sobelxy = cv2.Sobel(src=img\_blur, ddepth=cv2.CV\_64F, dx=1, dy=1, ksize=5)

#sobelxy = cv2.imshow('sobelxy', sobelxy)

var = np.var(norm\_image,axis=None)

sk = skew(norm\_image, axis=None)

kur = kurtosis(norm\_image, axis=None)

ent = entropy(norm\_image, axis=None)

ent = ent/100

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler(feature\_range=(0, 1))

result = clf.predict(np.array([[-0.91318,-2.0113,-0.19565,0.066365]]))

result = clf.predict(np.array([[var,sk,kur,ent]]))

print(result)

out = ""

if result[0] ==0:

out = "Real Currency"

else:

out = "Fake Currency"

######

fig = plt.figure(figsize=(3, 3))

plt.axis('off')

plt.imshow(stringToImage(my\_uploaded\_file\_base64))

imagedata = StringIO()

fig.savefig(imagedata, format='svg')

imagedata.seek(0)

imagedata.getvalue()

fig2 = plt.figure(figsize=(3, 3))

plt.axis('off')

plt.imshow(stringToEdgeImage(my\_uploaded\_file\_base64))

imagedata2 = StringIO()

fig2.savefig(imagedata2, format='svg')

imagedata2.seek(0)

imagedata2.getvalue()

if my\_uploaded\_file\_base64 != None:

return render(request, "result.html",{'original\_image':imagedata.getvalue(),

'edge\_image':imagedata2.getvalue(),

'variance':"{:.2f}".format(var),

'skew':"{:.2f}".format(sk),

'kurtosis':"{:.2f}".format(kur),

'entropy':"{:.2f}".format(ent),

'accuracy':accuracy,

'result':result,

'out':out})

except:

print("Notes picture not loaded")

return render(request, "index.html")

def result(request):

return render(request, "result.html")

1. **HTML Detection Panel Code**

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KyZXEAg3QhqLMpG8r+8fhAXLRk2vvoC2f3B09zVXn8CA5QIVfZOJ3BCsw2P0p/We" crossorigin="anonymous">

<title>Home</title>

{% load static %}

</head>

<nav class="navbar navbar-dark bg-dark justify-content-center">

<div class="container">

<a class="navbar-brand" href="/">

<img src="{% static 'myapp/icon.png' %}" alt="S" width="48" height="48">

Fake Currency Detector

</a>

</div>

</nav>

{% load static %}

<body background="{% static 'myapp/back13.png' %}" style="background-size: cover;background-repeat: no-repeat;background-attachment: fixed;">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-U1DAWAznBHeqEIlVSCgzq+c9gqGAJn5c/t99JyeKa9xxaYpSvHU5awsuZVVFIhvj" crossorigin="anonymous"></script>

<br><br>

<section class="vh-100 gradient-custom">

<div class="container py-4 h-90">

<div class="row d-flex justify-content-center align-items-center h-100">

<div class="col-12 col-md-8 col-lg-6 col-xl-5">

<div class="card bg-dark text-white" style="border-radius: 1rem;">

<div class="card-body p-3 text-center">

<div class="mb-md-5 mt-md-4 pb-5">

<h2 class="fw-bold mb-1 text-uppercase">Select Notes</h2>

<p class="text-white-50 mb-5">Please select note picture by click on the below button!</p>

<form name="voterform" method="POST" enctype="multipart/form-data">

{% csrf\_token %}

<div class="form-outline form-white mb-4">

<input type="file" id="my\_uploaded\_file" name="my\_uploaded\_file" accept="image/\*">

</div>

<br>

<button class="btn btn-outline-light btn-lg px-5" name="login" type="submit">Check Now</button>

</form>

</div>

</div>

</div>

</div>

</div>

</div>

</section>

</body>

</html>

**CHAPTER 5**

TESTING

## Testing Phase

Testing refers to test the software so it is also called software testing. **Software testing** is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test.[[1]](https://en.wikipedia.org/wiki/Software_testing#cite_note-Kaner_1-1) Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use.

Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test-

* meets the requirements that guided its design and development,
* responds correctly to all kinds of inputs,
* performs its functions within an acceptable time
* is sufficiently usable
* can be installed and run in its intended environments, and
* achieves the general result its stakeholders desire.

## 5.1 Unit Testing

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.

Parameterized unit tests (PUTs) are tests that take parameters. Unlike traditional unit tests, which are usually closed methods, PUTs take any set of parameters. PUTs have been supported by Testing, JUnit and various .NET test frameworks.

Suitable parameters for the unit tests may be supplied manually or in some cases are automatically generated by the test framework. Testing tools like Quick Check exist to generate test inputs for PUTs.

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits.

In our tool we tested each and every unit (or module) and it was successfully executed.

All units are working properly and perform their operation perfectly as expected.

## Integration Testing

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing.

Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

Some different types of integration testing are big-bang, mixed (sandwich), risky- hardest, top-down, and bottom-up. Other Integration Patterns are: collaboration integration, backbone integration, layer integration, client-server integration, distributed services integration and high-frequency integration.

In the big-bang approach, most of the developed modules are coupled together to form a complete software system or major part of the system and then used for integration testing. This method is very effective for saving time in the integration testing process.

However, if the test cases and their results are not recorded properly, the entire integration process will be more complicated and may prevent the testing team from achieving the goal of integration testing.

In our proposed system we tested all components by merge together and it passed all our criteria.

## System Testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic.

As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system.

The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

System testing is performed on the entire system in the context of a Functional Requirement Specification and/or a System Requirement Specification (SRS). System testing tests not only the design, but also the behaviour and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification.

After performing system testing with our currency detection system we find that it follows all requirements and working as per requirement, it takes the input as currency notes and after apply algorithm it give the proper result.

## Functional Testing

Functional testing is a [quality assurance](https://en.wikipedia.org/wiki/Quality_assurance) (QA) process and a type of [black-box testing](https://en.wikipedia.org/wiki/Black-box_testing) that bases its test cases on the specifications of the software component under test.

Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (unlike [white-box testing](https://en.wikipedia.org/wiki/White-box_testing)). Functional testing is conducted to evaluate the compliance of a system or component with specified functional [requirements](https://en.wikipedia.org/wiki/Requirement). Functional testing usually describes what the system does.

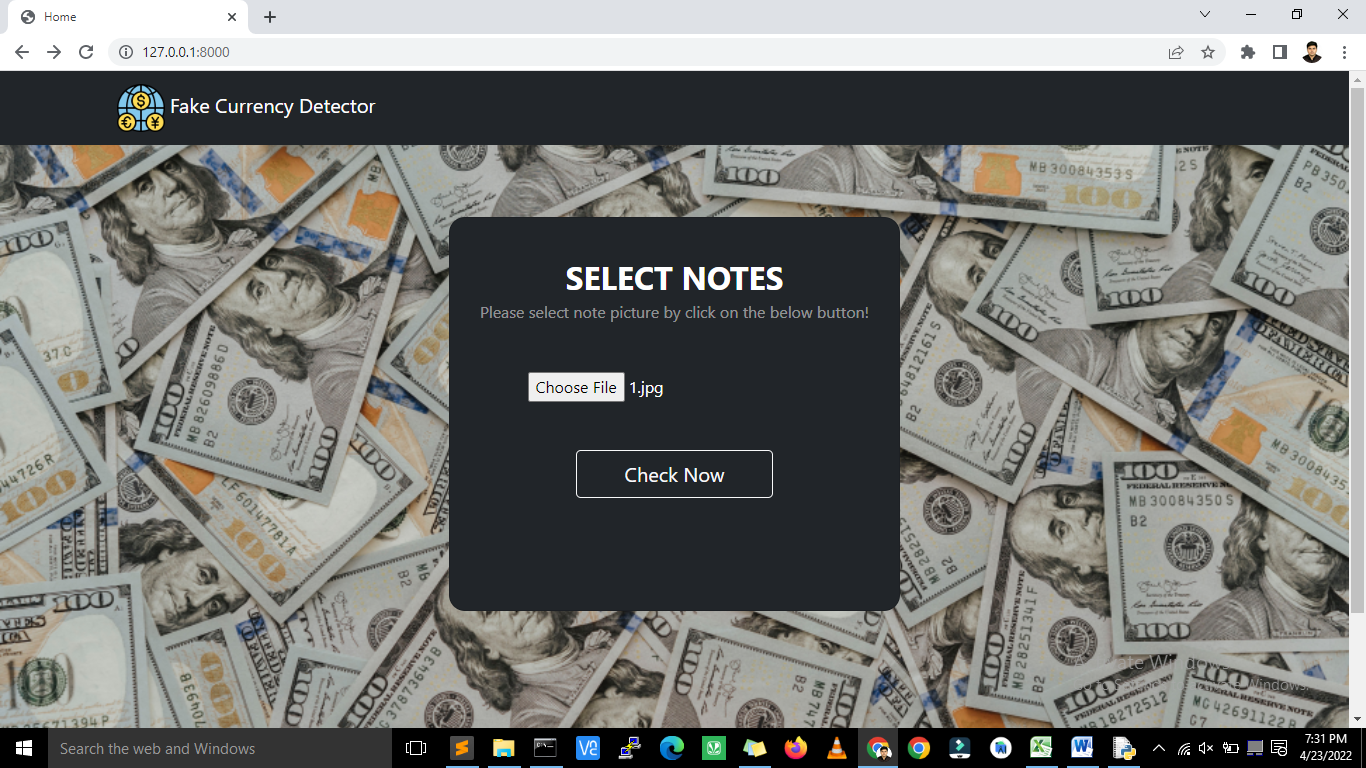
All the functionality of the currency detection system is working properly as expected.

It takes the input properly and after apply image processing generate the output as fake or real note.

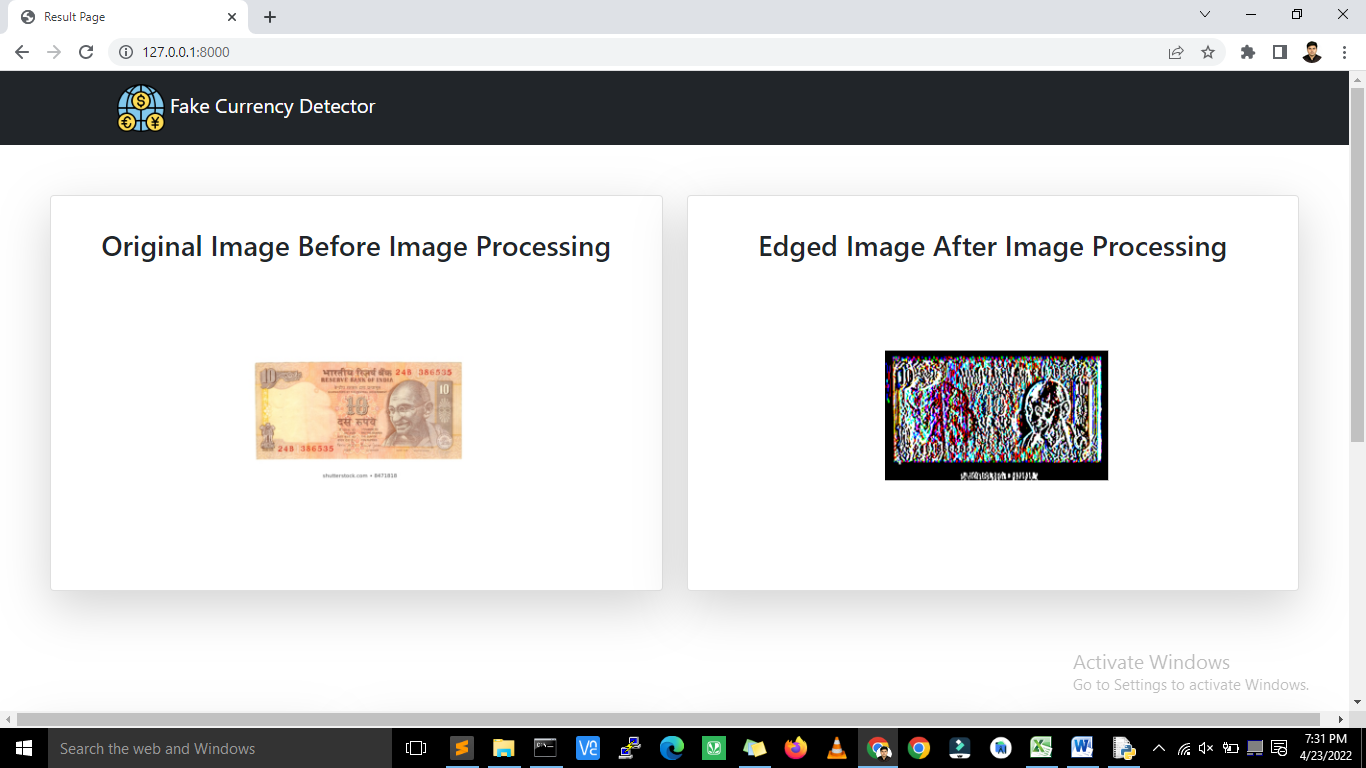
**CHAPTER 6**

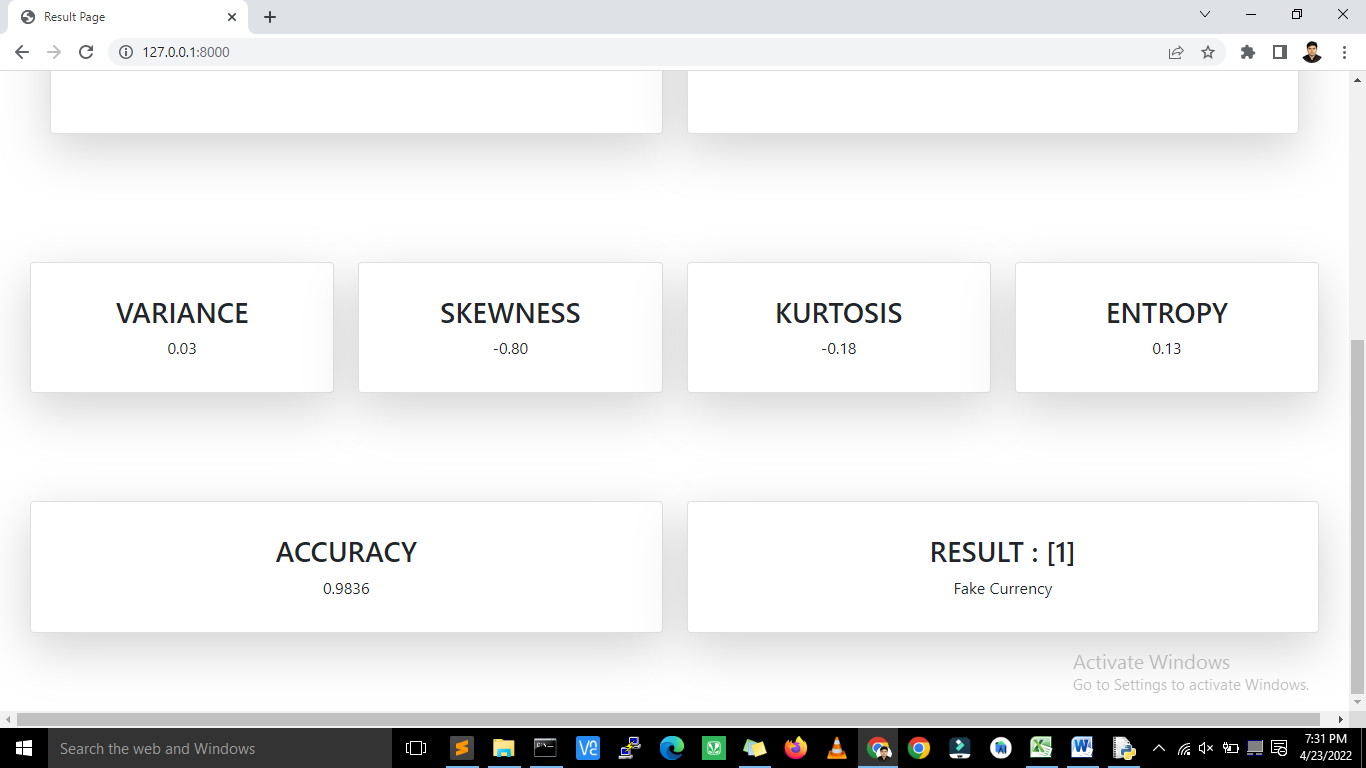
SNAPSHOTS

1. **Dashboard Page**

****

1. **Output Page**



****

**CHAPTER 7**

FUTURE ENHANCEMENT

**Future Enhancement**

In Future we will add some more features to our QR based attendance system like-

1. We can analyse the note based on the other features which will improve the accuracy of the system.
2. We will make this application live so that it can be easily accessible.
3. We will make the mobile application of the system so that mobile users can also use this application.
4. We will make the system more reliable and efficient so that it can solve the real world problem.
5. We will make API of our system so that we can use in the other system also.

**CONCLUSION**

The fake currency detection using image processing was implemented using Django framework to make the system easily accessible and image processing algorithms to identify the fake note. Features of currency note like variance, skew, kurtosis were extracted. The process starts from image acquisition to calculation of intensity of each extracted feature.

The algorithm processed here works suitably for the different currency notes. Hardware implementation of the proposed system can also be done using suitable processor so that to increase the speed of detection.

In this way our proposed system can eliminate the fake notes from the market and solve the real world problem.

**REFERENCES**

#### BOOKS:

* Ian Somerville ‘Software engineering’
* Rajeev mall ‘Software engineering’

#### ONLINE REFERENCE:

1. [www.google.co.in](http://www.google.co.in)
2. <https://stackoverflow.com/>
3. <https://docs.djangoproject.com/en/3.2/>
4. <https://www.w3schools.com/>