DAYANANDA SAGAR UNIVERSITY

KUDLU GATE, BENGALURU – 560068

Bachelor of Technology
in
COMPUTER SCIENCE AND ENGINEERING

Major Project Phase-II Report

PATIENT IMAGE RECOGNITION AND MEDICINE DISPENSING SYSTEM USING MACHINE LEARNING AND INTERNET OF THINGS (IoT)

By

Nobin Johnson – ENG18CS0197
Polaki Monika – ENG18CS0199
Prajwal S – ENG18CS0211
Thimmareddygari Jagadeeswar Reddy – ENG18CS0300

Under the supervision of

Dr. Basavaraj N Hiremath
Professor & Final Year B. Tech Co-Ordinator,
Department of Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING,
SCHOOL OF ENGINEERING
DAYANANDA SAGAR UNIVERSITY.

(2021-2022)

DAYANANDA SAGAR UNIVERSITY

School of Engineering
Department of Computer Science & Engineering
Kudlu Gate, Bengaluru – 560068
Karnataka, India

CERTIFICATE

This is to certify that the Phase-II project work titled **"PATIENT IMAGE RECOGNITION AND MEDICINE DISPENSING SYSTEM USING MACHINE LEARNING AND INTERNET OF THINGS (IOT)"** is carried out by **Nobin Johnson (ENG18CS0197), Polaki Monika (ENG18CS0199), Prajwal S (ENG18CS0211), Thimmareddygari Jagadeeswar Reddy (ENG18CS0300),** bonafide students of Bachelor of Technology in Computer Science and Engineering at the School of Engineering, Dayananda Sagar University, Bengaluru in partial fulfillment for the award of degree in Bachelor of Technology in Computer Science and Engineering, during the year **2021-2022**.

Dr. Basavaraj N Hiremath	**Dr. Girisha G S**	**Dr. A Srinivas**
Professor Dept. of CSE, School of Engineering Dayananda Sagar University	Chairman CSE School of Engineering Dayananda Sagar University	Dean School of Engineering Dayananda Sagar University
Date:	Date:	Date:

Name of the Examiner	**Signature of Examiner**
1.	
2.	

DECLARATION

We, **Nobin Johnson (ENG18CS0197), Polaki Monika (ENG18CS0199), Prajwal S (ENG18CS0211), and Thimmareddygari Jagadeeswar Reddy (ENG18CS0300),** are students of eighth-semester B.Tech in **Computer Science and Engineering**, at School of Engineering, **Dayananda Sagar University**, hereby declare that the phase-II project titled **"Patient Image Recognition and Medicine Dispensing System Using Machine Learning and Internet of Things (IoT)"** has been carried out by us and submitted in partial fulfillment for the award of degree in **Bachelor of Technology in Computer Science and Engineering** during the academic year **2021-2022**.

Student **Signature**

Name: Nobin Johnson
USN: ENG18CS0197

Name: Polaki Monika
USN: ENG18CS0199

Name: Prajwal S
USN: ENG18CS0211

Name: Thimmareddygari Jagadeeswar Reddy
USN: ENG18CS0300

Place: Bengaluru

Date:

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to Dayananda Sagar University, School of Engineering for providing us with a great opportunity to pursue our Bachelor's degree in this institution.

We would like to thank **Dr. A Srinivas. Dean**, **School of Engineering & Technology**, **Dayananda Sagar University** for his constant encouragement and expert advice. It is a matter of immense pleasure to express our sincere thanks to **Dr. Girisha G S, Department Chairman**, **Computer Science and Engineering**, **Dayananda Sagar University,** for providing the right academic guidance that made our task possible.

We would like to thank our guide **Dr. Basavaraj N Hiremath, Professor**, **Dept. of Computer Science and Engineering**, **Dayananda Sagar University**, for sparing his valuable time to extend help in every step of our project work, which paved the way for smooth progress and the fruitful culmination of the project.

We would like to thank our Project Coordinators Dr. Meenakshi Malhotra and Dr. Bharanidharan N, and all the staff members of Computer Science and Engineering for their support.

We are also grateful to our family and friends who provided us with every requirement throughout the course. We would like to thank one and all who directly or indirectly helped us in the Project work.

TABLE OF CONTENTS

NOMENCLATURE USED

ML	Machine Learning
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
IoT	Internet Of Things
JSON	JavaScript Object Notation
UI	User Interface
GPIO	General Purpose Input Output
IR	InfraRed
KNN	K Nearest Neighbors
BGR	Blue, Green, Red
RGB	Red, Green, Blue

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This project attempts to develop an Automated Medicine Dispensing System. Some of the modules involved are face recognition, the use of motors and detectors with IoT (Internet of Things), Database Management, and Web development.

An automated Medication Dispensing System detects the patient using unique features such as face identification. The system then fetches the medication prescribed to the identified patient from the database. The system then uses servo motors to dispense the tablet from a tube containing the tablets. When the patient reaches out their hand, the IR sensor detects the presence of an object, and the medication is dispensed. The quantity of the medication is updated in the database and displayed on a webpage in real-time. The system is user-friendly and efficient.

CHAPTER 1
INTRODUCTION

CHAPTER 1: INTRODUCTION

Due to the current state of the world, the medical industry has been burdened with the growth in the number of patients. This has led to excessive demand for faculties such as doctors and patients, which cannot perform the basic activities done before such as checkups and medication supply. The proposed system aims to automate these once performed routines such as medication dispensing using machine learning and Internet of Things (IoT) aspects.

1.1 SCOPE

In late 2019 a novel coronavirus, severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), caused an acute respiratory disease, coronavirus disease 2019 (covid-19), spread from its origins in China to become a pandemic. As of 26 March 2020, 455 770 cases had been identified worldwide, causing 20 740 deaths. No successful therapeutic intervention for covid-19 has yet been established, so supportive care is the most important aspect of clinical management, supporting the patient's physiology to aid recovery. Optimal provision of supportive care is therefore fundamental both to the wellbeing of individual patients and to securing the confidence of the general population. To enable the provision of the best care, clinicians need evidence-based recommendations developed using accepted methods. Such clinical guidelines must be readily available; hence this project is useful as a supportive care automated method. It's very useful in medical institutions and can be installed as a standard. Since it's scalable, a single unit can be provided for any number of patients. It will have a social impact due to the current state of the world. The contactless machinery in healthcare has grown in demand and this will be used to reduce the spreading of diseases as well as provide a sanitized and hygienic way of medication dispense.

CHAPTER 2

PROBLEM DEFINITION

CHAPTER 2: PROBLEM DEFINITION

To build an automated medication dispensing system that includes the Internet of Things (IoT) for motor and sensor controls which can address the challenge of the need for independence in supporting patients with less ability to identify and prescribe medicine and to maintain/ schedule those in the treatment. Machine Learning models will be implemented for facial recognition and person recognition systems using computer visions. Database and web development aspects will be integrated to build a visual tracker for the stock of medication remaining and a faculty portal for faculty operations through ethernet using database and HTML, CSS, and backend elements.

CHAPTER 3
LITERATURE REVIEW

CHAPTER 3: LITERATURE REVIEW

An elaborate literature survey was done for the relevant research. The following table contains a detailed summary of the papers, we used for our project.

Table 3.1 Literature Review

Si. No.	Title	Author	Description	Year
1	Path follower & obstacle avoidance smart robot vehicle: Applications of arduino programming	K. Ayush and R. Biswas	Working on Arduino and its application and design in the development of a path-following, obstacle avoidance robot.	2021
2	Servo Motor	H. Fattah	Describes the working of servo motor and stepper motor.	2021
3	Infrared sensors and ultrasonic sensors	X. Bi	Gives insights into the concepts of infrared and ultrasonic sensors.	2021
4	NoSQL comparison 2021: Couchbase server, MongoDB, and Cassandra (DataStax)	A. Yudovin and C. Gutierrez	Comparison and analysis between MongoDB and Cassandra.	2021
5	RT-MongoDB: A NoSQL database with differentiated performance	R. Andreoli, T. Cucinotta, and D. Pedreschi	Provides an extension to MongoDB that enables differentiated per-user/request performance.	2021

6	Accuracy enhancement in face recognition using 1D-PCA & 2D-PCA based on multilevel reverse-biorthogonal wavelet transform with KNN classifier	I. Dinariyah and Alamsyah	Implementation of decomposition and improving the accuracy of facial recognition in the AT&T and YALE dataset.	2021
7	An ingenious face recognition system based on HRPSM_CNN under unrestrained environmental condition	M. Tamilselvi and S. Karthikeyan	Implementation of HRPSM-CNN algorithm in the visually impaired assistive device, providing an accuracy rate of 97% & 96% for ORL, AR and LFW face database respectively.	2021
8	Machine Learning based Patient Face Recognition and Medicine Dispensing System using Raspberry Pi	Azher, Mohammed & Jyothi, K	Gives the basic architecture and flow chart of the system.	2020
9	Raspberry Pi	J. F. Nusairat	Revolves around coding to the Raspberry Pi device. Gives insight into how a connected environment can be built between cloud, device, and user.	2020

10	Performance analysis of NoSQL and relational databases with CouchDB and MySQL for application's data storage	C. A. Győrödi, D. V. Dumşe-Burescu, D. R. Zmaranda, R. Ş. Győrödi, G. A. Gabor, and G. D. Pecherle	Gives the advantages and disadvantages of each DBMS and is outlined in form of a performance comparison.	2020
11	An improved face recognition algorithm and its application in attendance management system	S. M. Bah and F. Ming	Presents a method known as the LBP algorithm combined with image processing techniques to address issues hampering face recognition.	2020
12	Face recognition systems: A survey	Y. Kortli, M. Jridi, A. A. Falou, and M. Atri	Reviews well-known techniques for facial recognition and analyses them.	2020
13	A Comparison of Arduino, Raspberry Pi and ESP8266 Boards	Ooko, Samson	Gives a detailed description between Arduino, Raspberry Pi, and ESP8266 boards.	2019
14	A review of comparison between NoSQL	Kaur, R. & Sahiwal, J.K.	Describes the comparison between MongoDB and CouchDB.	2019

	databases: MongoDB and couchDB			
15	A Review on Various Aspects of MongoDb Databases	A. Chauhan	Detailed analysis of MongoDB; giving insights into its advantages, uses issues, etc.	2019
16	Using modern web frameworks when developing an education application: a practical approach	S. Ivanova and G. Georgiev	Gives an overview and comparison of existing web technologies and frameworks.	2019
17	Implementation of database using python flask framework: College database management system	N. Chauhan, M. Singh, A. Verma, A. Parasher, and G. Budhiraja	Describes usage of python flask framework.	2019

CHAPTER 4
PROJECT DESCRIPTION

CHAPTER 4: PROJECT DESCRIPTION

An automated Medication Dispensing System detects the patient using unique features such as face identification. The system then fetches the medication prescribed to the identified patient from the database. The system then uses servo motors to dispense the tablet from a tube containing the tablets. When the patient reaches out their hand, the IR sensor detects the presence of an object, and the medication is dispensed. The quantity of the medication is updated in the database and displayed on a webpage in real-time. The system is user-friendly and efficient.

4.1 PROPOSED DESIGN

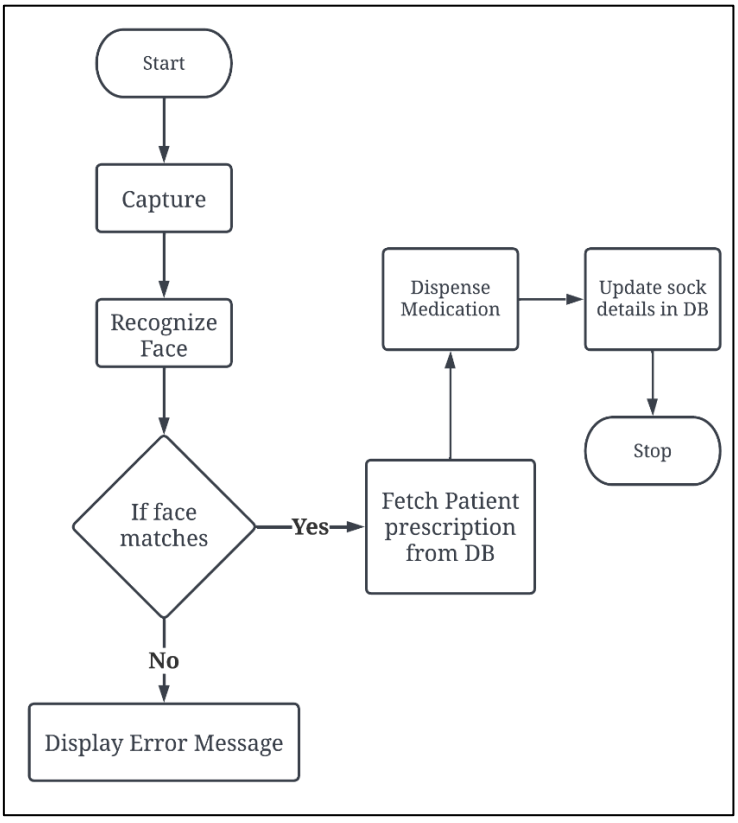


Figure 4.1.1 Flowchart of the dispensing system

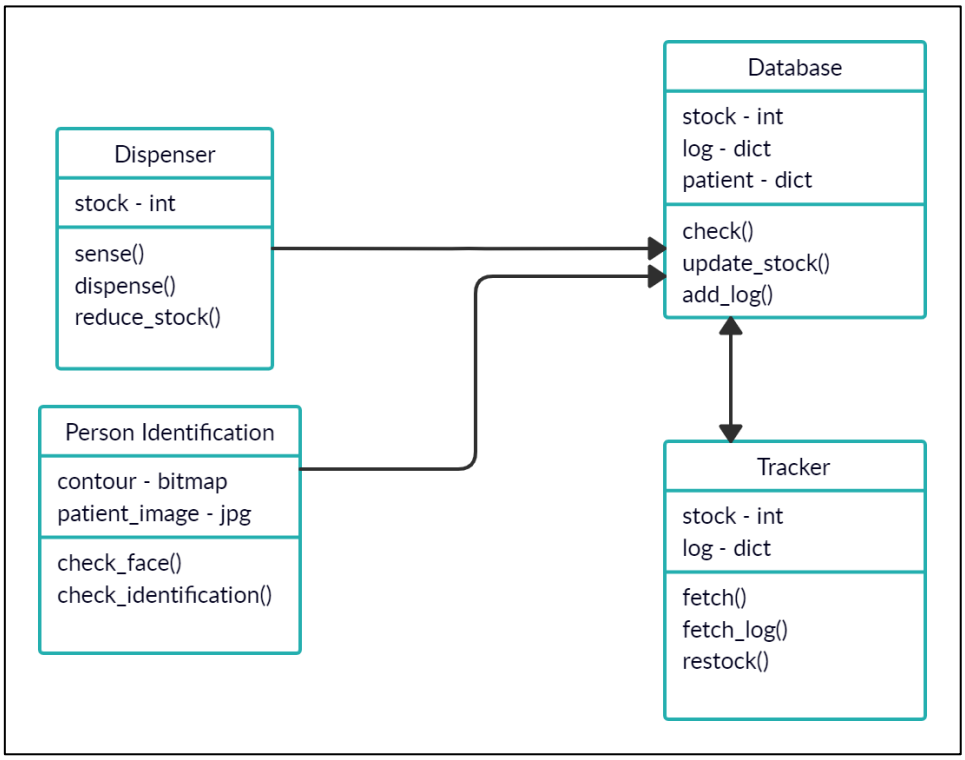


Figure 4.1.2 UML-Class diagram

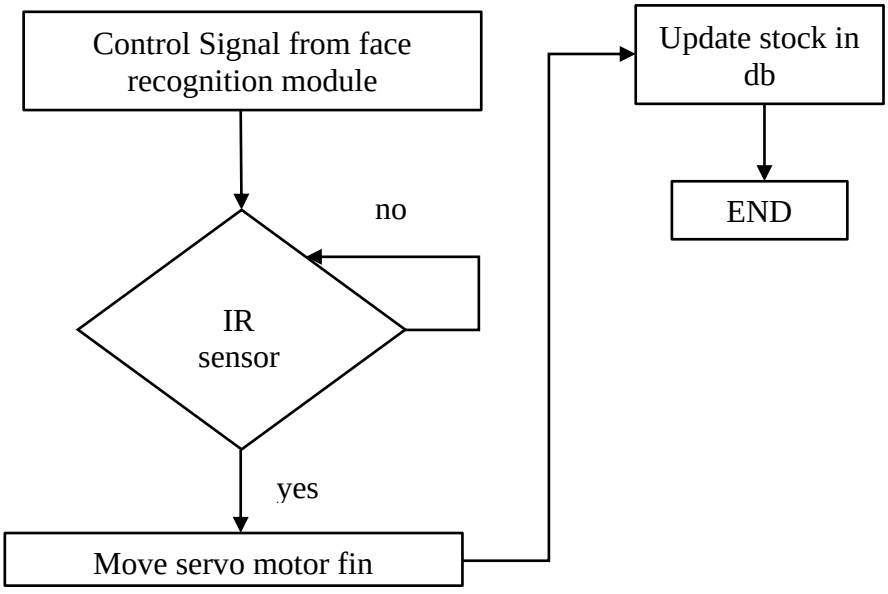


Figure 4.1.3 Dispensing module data flow diagram

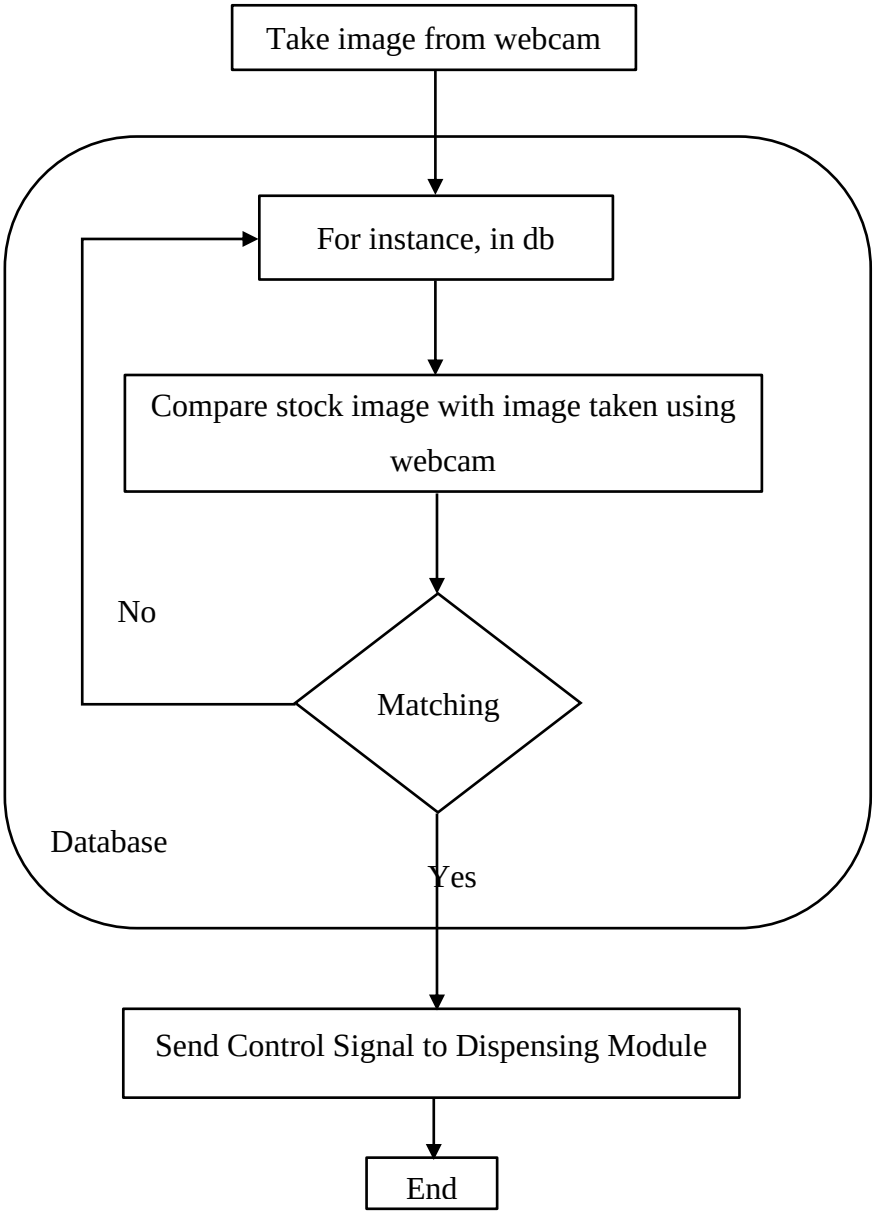


Figure 4.1.4 Face recognition module data flow diagram

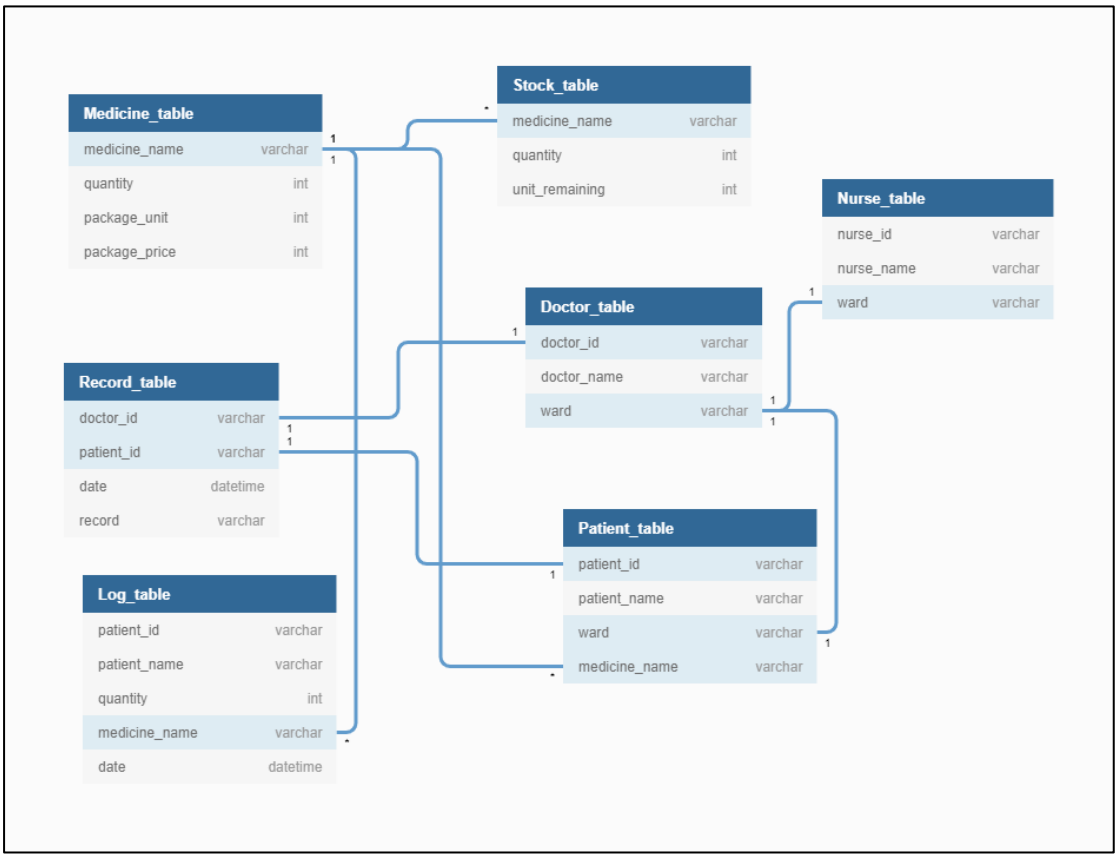


Figure 4.1.5- Database schema design

4.2 ASSUMPTIONS AND DEPENDENCIES

4.2.1. Assumptions

- Properly maintained database
- Proper lighting conditions

4.2.2. Dependencies of the project

- MongoDB
- Pymongo MongoDB python driver
- OpenCV python computer vision driver

- Flask python micro web framework
- RPi python raspberry Pi GPIO interface
- Nodejs
- Reactjs

CHAPTER 5
REQUIREMENTS

CHAPTER 5: REQUIREMENTS

5.1 FUNCTIONAL REQUIREMENTS

- Image Recognition – The system must be able to recognize the live video stream of the camera module using machine learning. Failure may result in wrong or non-dispensing.
- Image Processing – The system must be able to process images using computer vision and image processing embedded modules as well as dynamically process live streams.
- Real-Time system – The system must be able to work with real-time objects and supply the required output with minimal delay.

5.2 NON-FUNCTIONAL REQUIREMENTS

- Performance- The machine learning model used for image classification needs to have high accuracy.
- Availability – The system must be highly available as patients might need to use it multiple times throughout the day.
- Reliability – The system must be highly reliable.
- Portability – The system must be able to be ported to other systems containing a python environment.
- Scalability – The system must be highly scalable.

5.3 SOFTWARE REQUIREMENTS

Name	Version	Description
Python Ide Such as Visual Studio Code/Spyder/Google Collab JavaScript IDE such as Visual Studio Code or Atom	NA	Environment for writing and executing python and JavaScript programs
Computer vision(cv2)	4.5.3.56	Allows python programs to use cameras and image/video feeds
pymongo	3.12.0	Driver for MongoDB python3 interface
Hypertext Markup Language (HTML)	5	Used for creating the front-end structure of output webpage
Cascading Style Sheets (CSS)	3	Used to style the front end of the output webpage
JavaScript	ES2015	Used to script the front-end webpage
MongoDB	4.4	Used as the database to hold patient credentials
RPi.GPIO	0.7.0	Used to interface the gpio pins present in the Raspberry Pi using python
flask	2.0.x	Used as server program for output webpage

python	3.9.x	Used as the language to build machine learning models
Debian OS	11	Used as the primary operating system for Raspberry Pi
Virtual Network Computing(vnc) server	6.7.4	Used to access the Raspberry Pi UI
PuTTY	0.74	Used to access the Raspberry PI using ssh
IDE which contains JavaScript Runtime Environment such as Visual Studio Code or Code Sandbox	NA	JavaScript run time environment
NodeJS	17.3.0	Used for server-side programming and it is built on chrome v8 JavaScript engine
ReactJS	17.0.1	Used to build user interfaces for web application

5.4 HARDWARE REQUIREMENTS

- Raspberry Pi 2/3/4
- Ethernet Cable
- Micro SD Card
- Router with port forwarding option
- 2-ampere power supply

- Micro USB cable
- USB web camera (Logitech c270 HD camera)
- Servo motors
- Connecting Wires

CHAPTER 6
METHODOLOGY

CHAPTER 6: METHODOLOGY

The Raspberry pi will be used as the microcontroller due to its portable, volatile, and user-friendly nature. The IR sensor and servo motors will be used in conjunction due to their specific features which serve as requirements in this system. The best database for our particular system will be MongoDB as the webpage is a simplistic structure and MongoDB offers fast connection and operations with a NoSQL and documented oriented data structure. The CNN pre-trained model and the SVM is the best-suited algorithm for facial recognition systems and it is much more efficient and sophisticated. The Backend for the tracker webpage will be created using Nodejs and express, due to its robust, minimal, and flexible set of features which is required for our single route webpage.

6.1 METHODS

- Face-Recognition - The patient's image is captured using an external webcam. It is then processed through a facial recognition system that detects and recognizes the face of the patient. Once the patient's face is recognized, a signal/call is sent to the medicine dispensing module.
- Medicine Dispensing - Medicine dispensing is an essential part of our project. The doctor includes the type of medicine that has to be given to the patient. Once the type of medicine is decided, the respective medicine is dispensed through the cylindrical container. IR sensor is used to monitor the dispensing of the medicine. The quantity of medicine dispensed is updated in real-time on the website as well as the database.
- Database – The database plays a crucial role in the development of this project. The details related to the doctor, nurse, ward, and patient are maintained.

Information about the medicine and also the stack images of faces are stored in the database. MongoDB is chosen as a database because of its robust nature, powerful querying, and analysis, etc.

- Website – The front end of the project, which houses the details related to medication dispensed, the quantity of medication left, etc. are built using HTML, CSS, JavaScript, and ReactJS. The doctor and nurse portal are present which helps the doctors and nurses to log in using their unique credentials and check the details about the patient, time of medication dispensed, etc.

6.2 DATA COLLECTION

- The primary data that is required is the images of the patient. Also, we require the basic details of the patient such as patient name, age, gender, dependent, address, etc. These details will be stored in our database.
- For facial recognition, an external webcam - a Logitech c270 HD camera is used.
- We are taking 1000 images of each team member to train the model.
- Sample prescription with medicine and quantity.

CHAPTER 7
EXPERIMENTATION

CHAPTER 7: EXPERIMENTATION

Problems or setbacks arise in almost every project/real-world problem. Adequate steps have to be taken to correct these problems and move ahead to find the solution. Even in our project, we faced a few setbacks.

The major problem that we faced while implementing the project is integrating the various modules. Communication between the modules was difficult to handle.

We solved this communication problem by making use of REST services. RESTful web services are REST architecture-based web services. In this architecture, everything is a resource. They are lightweight, highly scalable, and maintainable; and are very commonly used to create APIs for web-based applications.

The below lines of codes are the most prominent part of our project. They form the core of the project and will give us insights into the various modules and functions of the project.

7.1 DISPENSER FUNCTION

The dispenser function contains the lines of code required for dispensing the medication to the patient. A 5-second time frame is given between dispensing 2 or more medications so that it is easy for the patient to collect the medication from the medicine dispensing tube.

```
class Dispenser():
    def __init__(self,ir,servo,name):
        self.name=name
        self.ir=ir
        self.servo=servo
    def dispense(self):
        global duty
```

```
servo_output=GPIO.PWM(self.servo,50)
while True:
    ir_sensor=GPIO.input(self.ir)
    if ir_sensor==0:
        servo_output.start(5)
        while duty<8:
            servo_output.ChangeDutyCycle(duty)
            time.sleep(0.1)
            print(duty)
            duty=duty+1
        while duty>5:
            servo_output.ChangeDutyCycle(duty)
            time.sleep(0.1)
            duty=duty-1
            print(duty)
        number=Quantity.find_one({'name':self.name})['quantity']
        Quantity.delete_one({'name':self.name})
        Quantity.insert_one({'name':self.name,'quantity':number-1})
        Log.insert_one({'medicine':self.name,'time':str(datetime.datetime.now())})
        break;
```

7.2 SERVER CODE (BACKEND)

There are 6 tables in our database – stock table, record table, log table, doctor table, nurse table, and patient table. They store relevant information about the characteristic of each table.

```
Doctor=client.Doctor_Database
Nurse=client.Nurse_Database
Patient=client.Patient_Database
```

```
Log=client.Log_Database
Record=client.Record_Database
Stock=client.Stock_Database
Stock_table=Stock.Table
Record_table=Record.Table
Log_table=Log.Table
Doctor_table=Doctor.Table
Nurse_table=Nurse.Table
Patient_table=Patient.Table
```

The below lines of code explain the functions related to the fetching of the medicines A, B, C and D.

```
def fetchA():
  if Log_table.count_documents({'Medication':'Medicine 1'})==0:
      return jsonify({})
  else:
      d=[]
      for documents in Log_table.find({'Medication':'Medicine 1'}):

d.append({'Medication':documents['Medication'],'Patient':documents['Patient'],'Time':
documents['Time']})
      return jsonify(d)
@page.route('/fetchB',methods=['GET'])
def fetchB():
  if Log_table.count_documents({'Medication':'Medicine 2'})==0:
      return jsonify({})
  else:
      d=[]
```

```
        for documents in Log_table.find({'Medication':'Medicine 2'}):

d.append({'Medication':documents['Medication'],'Patient':documents['Patient'],'Time':
documents['Time']})
        return jsonify(d)
@page.route('/fetchC',methods=['GET'])
def fetchC():
  if Log_table.count_documents({'Medication':'Medicine 3'})==0:
        return jsonify({})
  else:
        d=[]
        for documents in Log_table.find({'Medication':'Medicine 3'}):

d.append({'Medication':documents['Medication'],'Patient':documents['Patient'],'Time':
documents['Time']})
        return jsonify(d)
@page.route('/fetchD',methods=['GET'])
def fetchD():
  if Log_table.count_documents({'Medication':'Medicine 4'})==0:
        return jsonify({})
  else:
        d=[]
        for documents in Log_table.find({'Medication':'Medicine 4'}):

d.append({'Medication':documents['Medication'],'Patient':documents['Patient'],'Time':
documents['Time']})
        return jsonify(d)
@page.route('/checkStock',methods=['GET'])
```

7.3 FRONT END CODE

The front end of the website is designed using HTML, CSS and ReactJS.

```
import './App.css';
import Navbars from './Screens/navbar';
import Home from './Screens/Home'
import {BrowserRouter as Router,Switch,Route} from "react-router-dom"
import 'bootstrap/dist/css/bootstrap.min.css';
import Doctor from './Screens/doctor-portal.js'
import Nurse from './Screens/Nurse.js'
import Footer from './Screens/Footer';
import LOGA from './Screens/logA'
import LOGB from './Screens/logb'
import LOGC from './Screens/logC'
import LOGD from './Screens/logD'
```

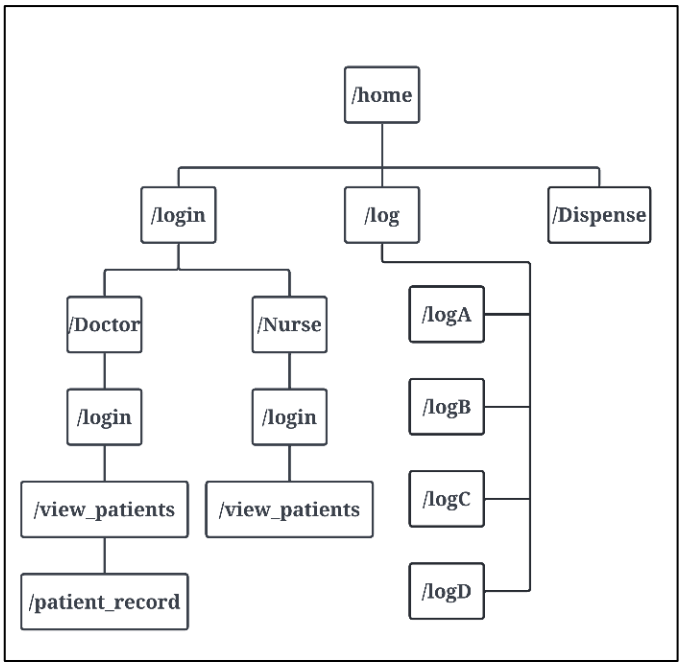


Figure 7.3.1: Directory and sub-directories of the front end.

7.4 FACE RECOGNITION

The face recognition module in our project plays an important role. It detects and recognizes the face. A deep learning pre-trained model, the Caffe Model, is used to detect faces. The bounding boxes are drawn and a blob is created from the image i.e., the image is ready in the input format for the neural network and performs the pre-processing on the image. The image is re-sized using mean subtraction and the image is converted from BGR to RGB format.

The below code extracts the features from the face using OpenCV.

```
for (i, imagePath) in enumerate(imagePaths):
# extract the person name from the image path
if (i%50 == 0):
    print("Processing image {}/{}".format(i, len(imagePaths)))
    # print(imagePaths)
name = imagePath.split(os.path.sep)[-2]

# load the image, resize it to have a width of 600 pixels (while maintaining the aspect ratio), and then grab the image dimensions
image = cv2.imread(imagePath)
image = imutils.resize(image, width=600)
(h, w) = image.shape[:2]

# construct a blob from the image
imageBlob = cv2.dnn.blobFromImage(
    cv2.resize(image, (300, 300)), 1.0, (300, 300),
    (104.0, 177.0, 123.0), swapRB=False, crop=False)

# apply OpenCV's deep learning-based face detector to localize faces in the input image
```

```
detector.setInput(imageBlob)
detections = detector.forward()
```

The below set of code detects and recognizes the face in the live video stream via the webcam.

```
for i in range(0, detections.shape[2]):
    # extract the confidence (i.e., probability) associated with the prediction
    confidence = detections[0, 0, i, 2]

    # filter out weak detections
    if confidence > 0.5:
        # compute the (x, y)-coordinates of the bounding box for the face
        box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
        (startX, startY, endX, endY) = box.astype("int")

        # extract the face ROI
        face = frame[startY:endY, startX:endX]
        (fH, fW) = face.shape[:2]

        # ensure the face width and height are sufficiently large
        if fW < 20 or fH < 20:
            continue

        # construct a blob for the face ROI, then pass the blob through our face
embedding model to obtain the 128-d quantification of the face
        faceBlob = cv2.dnn.blobFromImage(face, 1.0 / 255,
            (96, 96), (0, 0, 0), swapRB=True, crop=False)
        embedder.setInput(faceBlob)
```

```
vec = embedder.forward()

# perform classification to recognize the face
preds = recognizer.predict_proba(vec)[0]
j = np.argmax(preds)
proba = preds[j]
name = le.classes_[j]
```

CHAPTER 8
TESTING AND RESULTS

CHAPTER 8: TESTING AND RESULTS

The below table gives us insights into the test cases and the results of the testing that has been performed in this project.

Table 8.1 Analysis of test cases

TEST CASE	EXPECTED OUTPUT	RECORDED OUTPUT
FACE RECOGNITION		
No face in the camera	Stream continues till face is detected.	Stream continues till face is detected.
Patient Face	Dispense prescribed medicine.	Dispense prescribed medicine.
Unknown Face	Error message saying “Unknown”.	Error message saying “Unknown”.
DISPENSING UNIT		
Normal Case	Medication is dispensed.	Medication is dispensed.
One tablet remaining	Medication is dispensed and the stock light is turned on.	Medication is dispensed and the stock light is turned on.
Zero tablets remaining	Error message saying “Please restock medication”.	Error message saying “Please restock medication”.

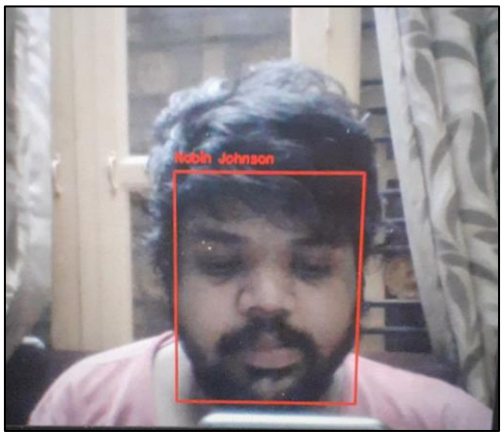


Figure 8.1: Test Case 1 - Expected and Predicted outputs are the same.

Expected Output: Nobin Johnson, Id - 1000

Predicted Output: Nobin Johnson, Id - 1000

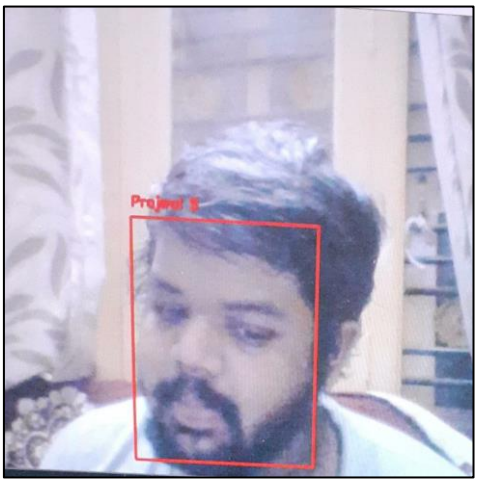


Figure 8.2: Test Case 2 - Expected output is different from the predicted output.

Expected Output: Nobin Johnson, Id -1000

Predicted Output: Prajwal S, Id -1002

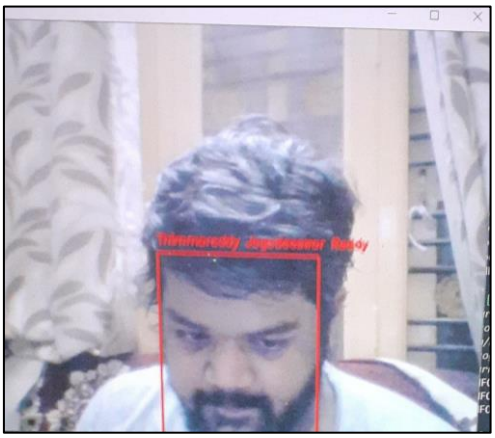


Figure 8.3: Test Case 3 - Expected output is different from the predicted output.

Expected Output: Nobin Johnson, Id -1000

Predicted Output: Thimmareddygari Jagadeeswar Reddy, Id -1003

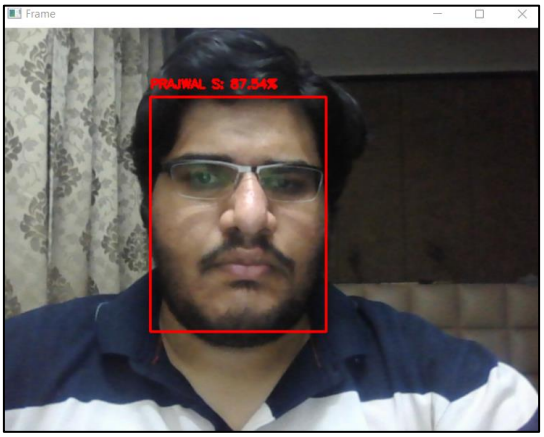


Figure 8.4: Test Case 4 - Expected and Predicted outputs are the same.

Expected Output: Prajwal S, Id -1002

Predicted Output: Prajwal S, Id -1002

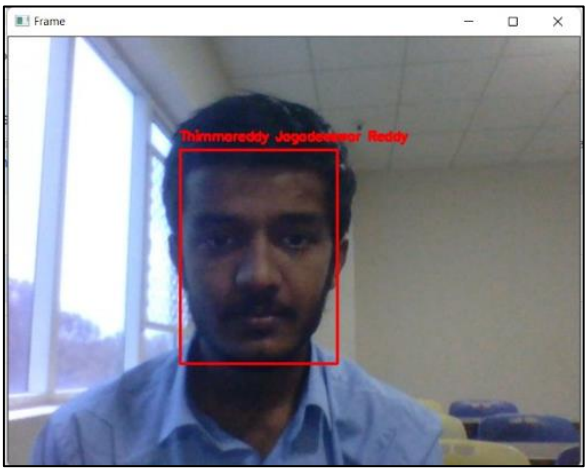


Figure 8.5: Test Case 5 - Expected and Predicted outputs are the same.

Expected Output: Thimmareddygari Jagadeeswar Reddy, Id -1003

Predicted Output: Thimmareddygari Jagadeeswar Reddy, Id -1003

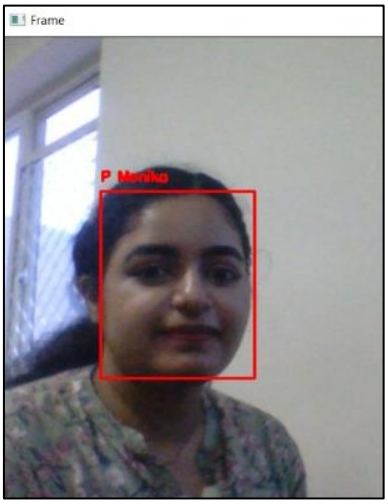


Figure 8.6: Test Case 6 - Expected and Predicted outputs are the same.

Expected Output: P Monika, Id -1001

Predicted Output: P Monika, Id - 1001

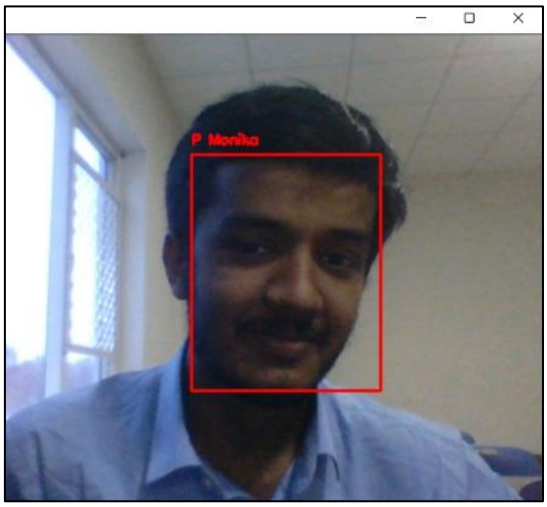


Figure 8.7: Test Case 7 - Expected and Predicted outputs are different.

Expected Output: Thimmareddygari Jagadeeswar Reddy, Id -1003

Predicted Output: P Monika, Id -1001

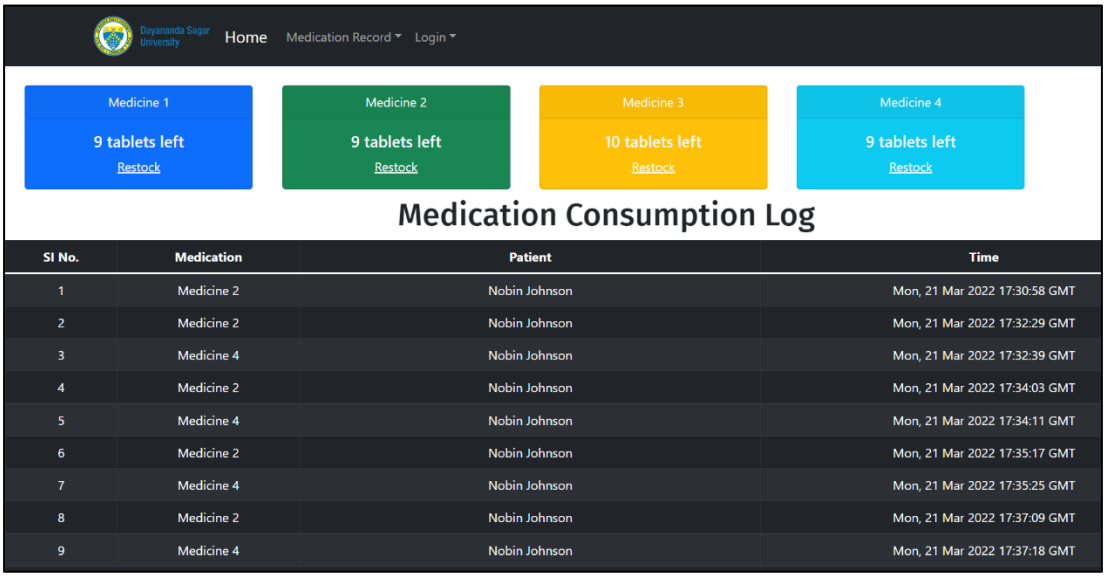
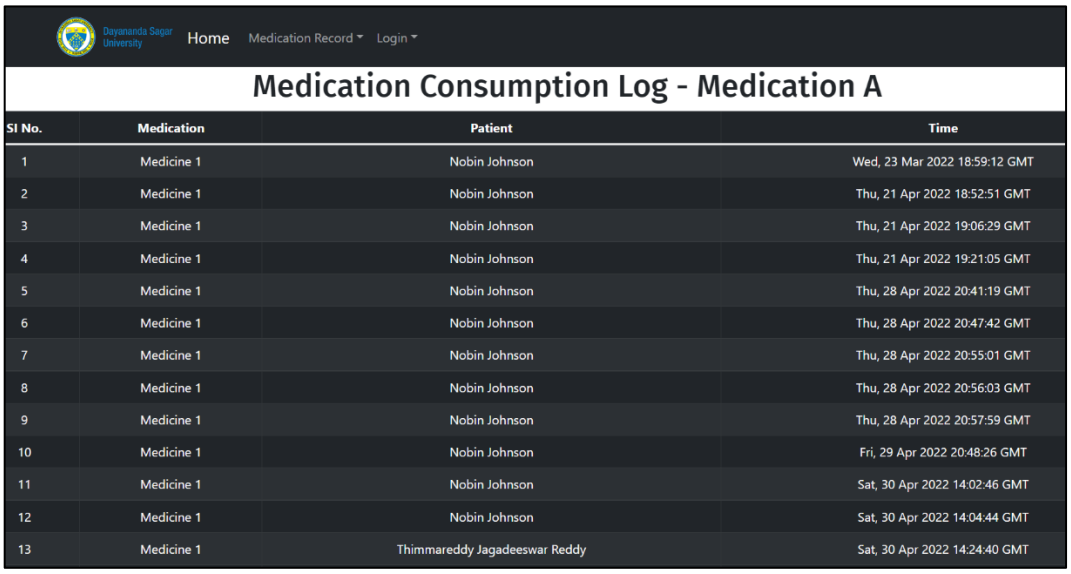


Figure 8.8: Homepage of the website.



Sl No.	Medication	Patient	Time
1	Medicine 1	Nobin Johnson	Wed, 23 Mar 2022 18:59:12 GMT
2	Medicine 1	Nobin Johnson	Thu, 21 Apr 2022 18:52:51 GMT
3	Medicine 1	Nobin Johnson	Thu, 21 Apr 2022 19:06:29 GMT
4	Medicine 1	Nobin Johnson	Thu, 21 Apr 2022 19:21:05 GMT
5	Medicine 1	Nobin Johnson	Thu, 28 Apr 2022 20:41:19 GMT
6	Medicine 1	Nobin Johnson	Thu, 28 Apr 2022 20:47:42 GMT
7	Medicine 1	Nobin Johnson	Thu, 28 Apr 2022 20:55:01 GMT
8	Medicine 1	Nobin Johnson	Thu, 28 Apr 2022 20:56:03 GMT
9	Medicine 1	Nobin Johnson	Thu, 28 Apr 2022 20:57:59 GMT
10	Medicine 1	Nobin Johnson	Fri, 29 Apr 2022 20:48:26 GMT
11	Medicine 1	Nobin Johnson	Sat, 30 Apr 2022 14:02:46 GMT
12	Medicine 1	Nobin Johnson	Sat, 30 Apr 2022 14:04:44 GMT
13	Medicine 1	Thimmareddy Jagadeeswar Reddy	Sat, 30 Apr 2022 14:24:40 GMT

Figure 8.9: Log of the medicine A that has been dispensed to the patient.

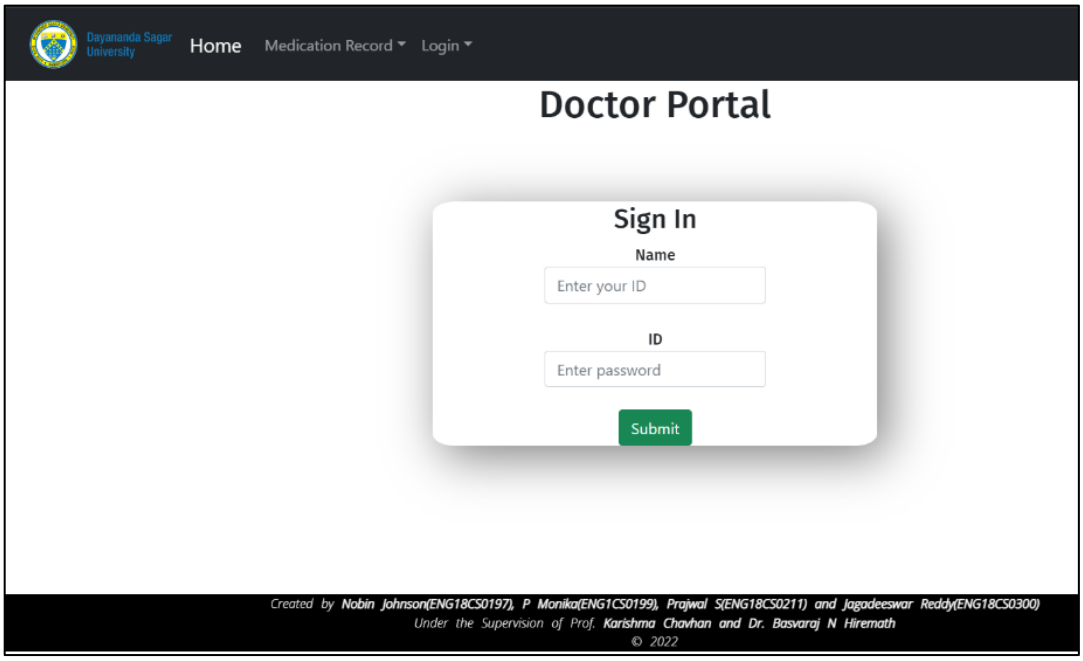


Figure 8.10: Sign in portal for the doctor.

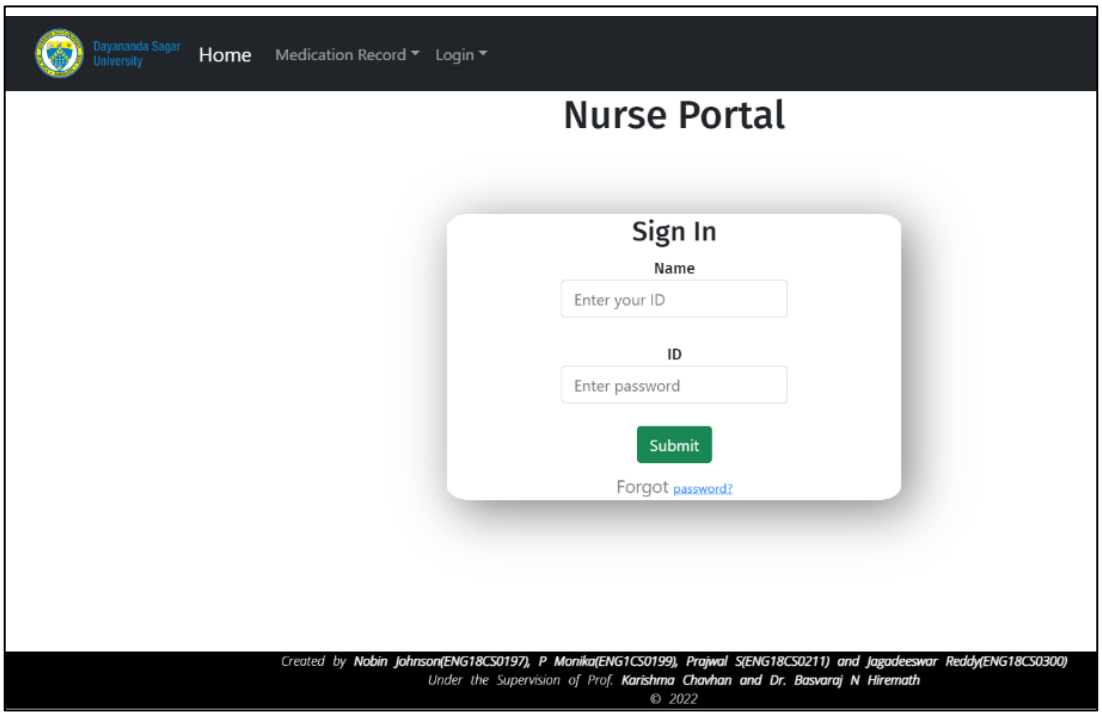


Figure 8.11: Sign in portal for the nurse.

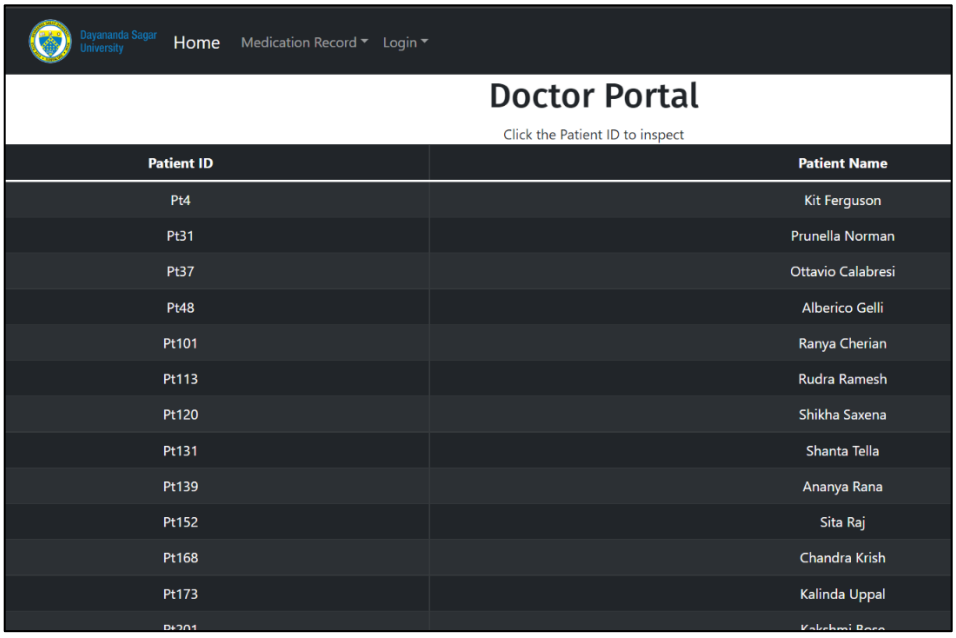


Figure 8.12: Details regarding the patient i.e., patient ID and patient name that is visible to the doctor, once the doctor has logged in.

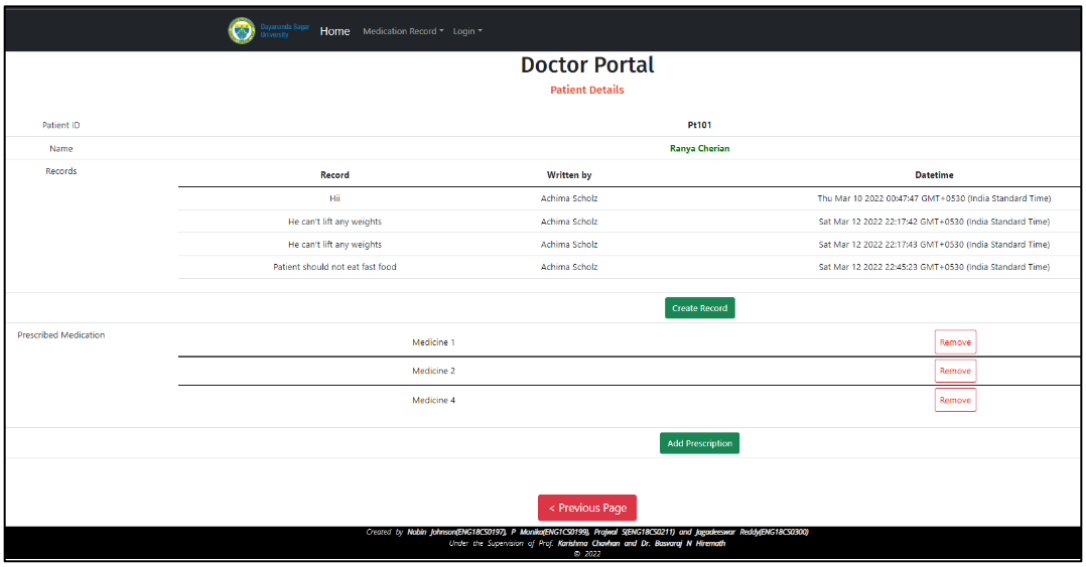
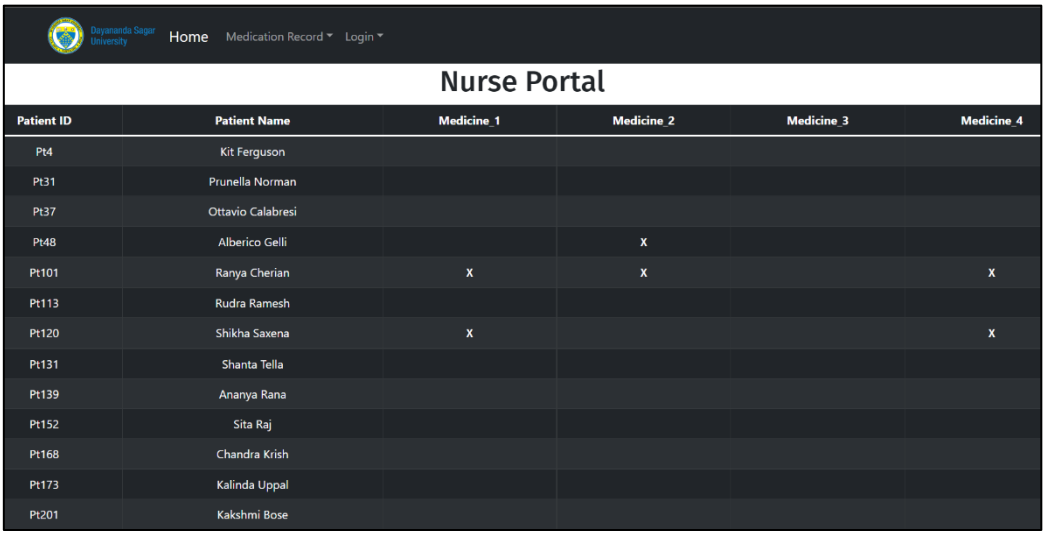


Figure 8.13: Details pertaining to the particular patient once the doctor has logged in and clicked on a particular patient ID.



Patient ID	Patient Name	Medicine_1	Medicine_2	Medicine_3	Medicine_4
Pt4	Kit Ferguson				
Pt31	Prunella Norman				
Pt37	Ottavio Calabresi				
Pt48	Alberico Gelli		X		
Pt101	Ranya Cherian	X	X		X
Pt113	Rudra Ramesh				
Pt120	Shikha Saxena	X			X
Pt131	Shanta Tella				
Pt139	Ananya Rana				
Pt152	Sita Raj				
Pt168	Chandra Krish				
Pt173	Kalinda Uppal				
Pt201	Kakshmi Bose				

Figure 8.14: Medications that are prescribed to a patient is visible once the nurse has logged in.

CHAPTER 9

CONCLUSION AND FUTURE WORK

CHAPTER 9: CONCLUSION AND FUTURE WORK

The Automatic Medicine dispensing system makes use of the latest technologies and is ergonomic in design. This system can be incorporated into clinics, pharmacies, and laboratories. The system is scalable, portable, reliable, and provides high availability. The web page provides a UI that is user-friendly and easy to use. There is a lot of scope wherein, the system can be optimized further and more technical aspects can be added.

In future iterations, different aspects of the system may be improved. A different web development stack may be used for the webpage as well as the mobile application. Fetch API provides basic features and may be replaced with a more robust API. A different database that is small may be used instead of MongoDB such as Firebase or SQLite. The webpage may be hosted using on-premise servers instead of using cloud services.

The system can be updated where in the following scenarios can be considered: patient taking medicine at the wrong time instead of the prescribed time; different sized capsules have to be dispensed; integrating the system with a mobile app so that notifications are sent periodically to the patient to consume the medication and so on.

REFERENCES

REFERENCES

[1] K. Ayush and R. Biswas, “Path follower & obstacle avoidance smart robot vehicle: Applications of arduino programming,” Int. J. Appl. Eng. Res., vol. 16, no. 6, p. 509, 2021.

[2] H. Fattah, “Servo Motor,” in LTETM Cellular Narrowband Internet of Things (NB-IoT), CRC Press, 2021, pp. 319–324.

[3] X. Bi, “Infrared sensors and ultrasonic sensors,” in Environmental Perception Technology for Unmanned Systems, Singapore: Springer Singapore, 2021, pp. 143–168.

[4] A. Yudovin and C. Gutierrez, “NoSQL comparison 2021: Couchbase server, MongoDB, and Cassandra (DataStax),” Altoros.com, 30-Mar-2021. [Online]. Available: https://www.altoros.com/blog/nosql-comparison-2021-couchbase-server-mongodb-and-cassandra-datastax/. [Accessed: 09-Dec-2021].

[5] R. Andreoli, T. Cucinotta, and D. Pedreschi, “RT-MongoDB: A NoSQL database with differentiated performance,” in Proceedings of the 11th International Conference on Cloud Computing and Services Science, 2021.

[6] I. Dinariyah and Alamsyah, “Accuracy enhancement in face recognition using 1D-PCA & 2D-PCA based on multilevel reverse-biorthogonal wavelet transform with KNN classifier,” J. Phys. Conf. Ser., vol. 1918, no. 4, p. 042144, 2021.

[7] M. Tamilselvi and S. Karthikeyan, “An ingenious face recognition system based on HRPSM_CNN under unrestrained environmental condition,” Alex. Eng. J., 2021.

[8] Azher, Mohammed & Jyothi, K.. (2020). Machine Learning based Patient Face Recognition and Medicine Dispensing System using Raspberry Pi. IOP Conference Series: Materials Science and Engineering. 981. 042007. 10.1088/1757-899X/981/4/042007

[9] J. F. Nusairat, “Raspberry Pi,” in Rust for the IoT, Berkeley, CA: Apress, 2020, pp. 391–427.

[10] C. A. Győrödi, D. V. Dumşe-Burescu, D. R. Zmaranda, R. Ş. Győrödi, G. A. Gabor, and G. D. Pecherle, "Performance analysis of NoSQL and relational databases with CouchDB and MySQL for application's data storage," Appl. Sci. (Basel), vol. 10, no. 23, p. 8524, 2020.

[11] S. M. Bah and F. Ming, "An improved face recognition algorithm and its application in attendance management system," Array, vol. 5, no. 100014, p. 100014, 2020.

[12] Y. Kortli, M. Jridi, A. A. Falou, and M. Atri, "Face recognition systems: A survey," Sensors (Basel), vol. 20, no. 2, p. 342, 2020

[13] Ooko, Samson. (2019). A Comparison of Arduino, Raspberry Pi and ESP8266 Boards.

[14] Kaur, R. & Sahiwal, J.K.. (2019). A review of comparison between NoSQL databases: MongoDB and couchDB. International Journal of Recent Technology and Engineering. 7. 892-898.

[15] A. Chauhan, "A Review on Various Aspects of MongoDb Databases," Int. J. Eng. Res. Technol. (Ahmedabad), vol. 8, no. 5, 2019.

[16] S. Ivanova and G. Georgiev, "Using modern web frameworks when developing an education application: a practical approach," in 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2019

[17] N. Chauhan, M. Singh, A. Verma, A. Parasher, and G. Budhiraja, "Implementation of database using python flask framework: College database management system," Int. j. eng. comput. sci., vol. 8, no. 12, pp. 24894–24899, 2019.

APPENDIX A

APPENDIX A

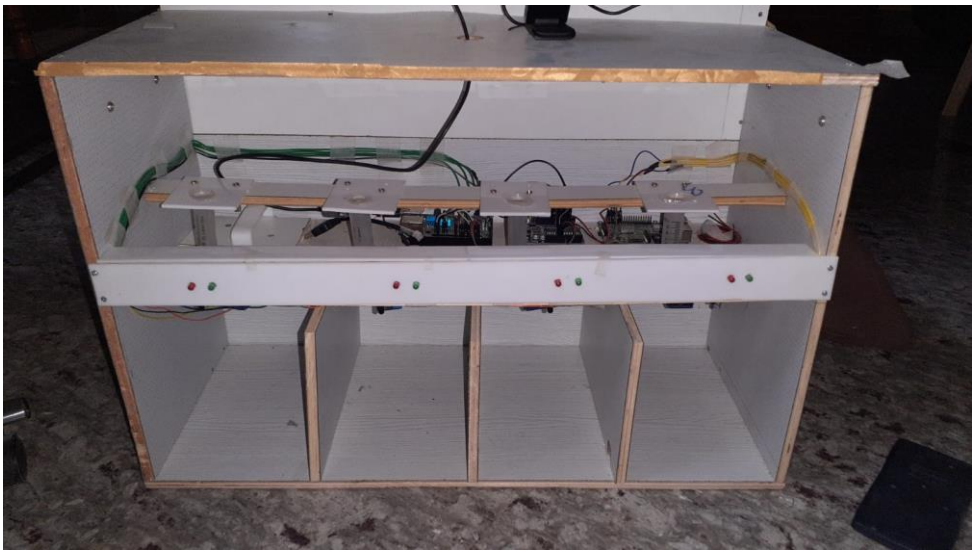


Figure A.1: Front view of the Medicine Dispensing System.

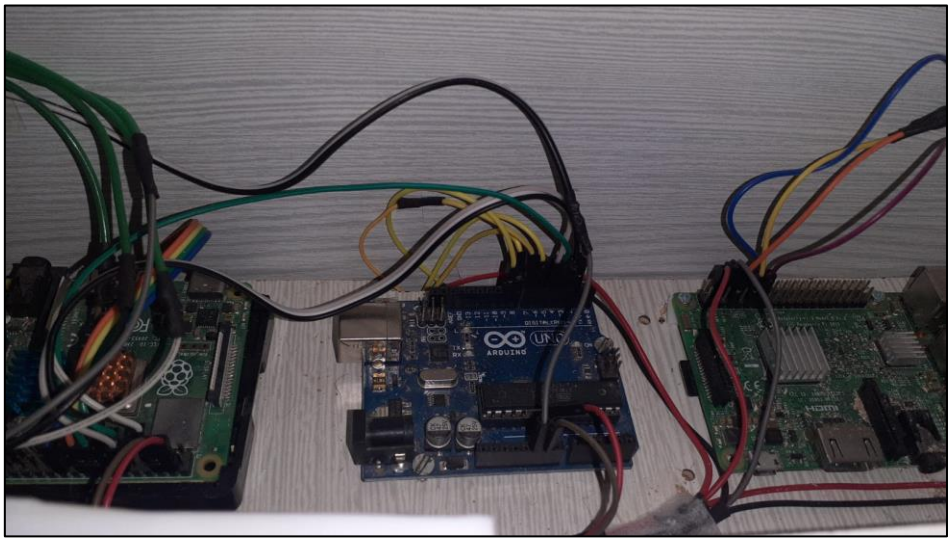


Figure A.2: Raspberry Pi Setup

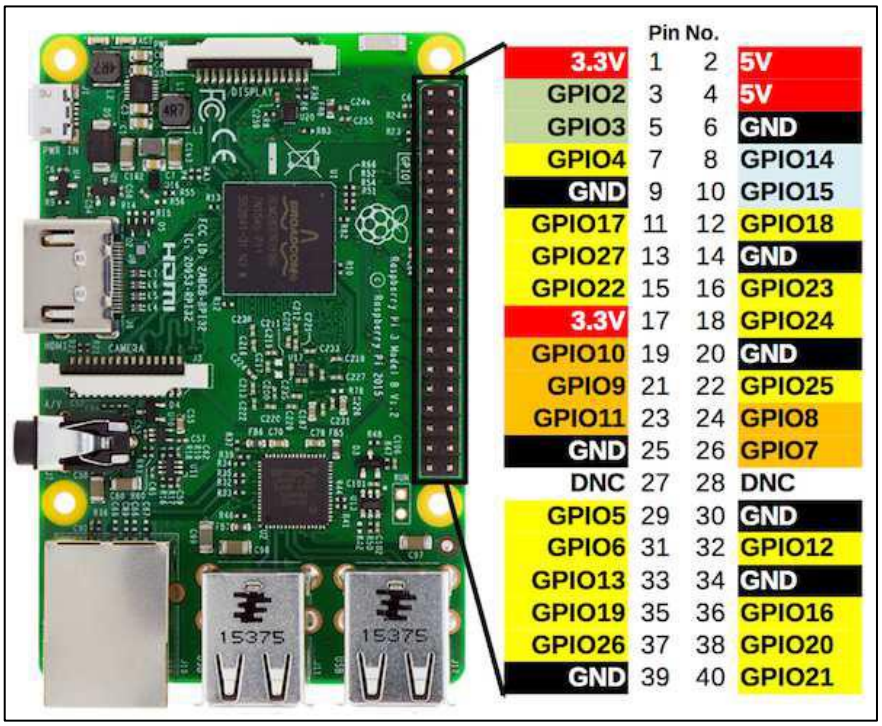


Figure A.3: Raspberry Pi 4B Architecture

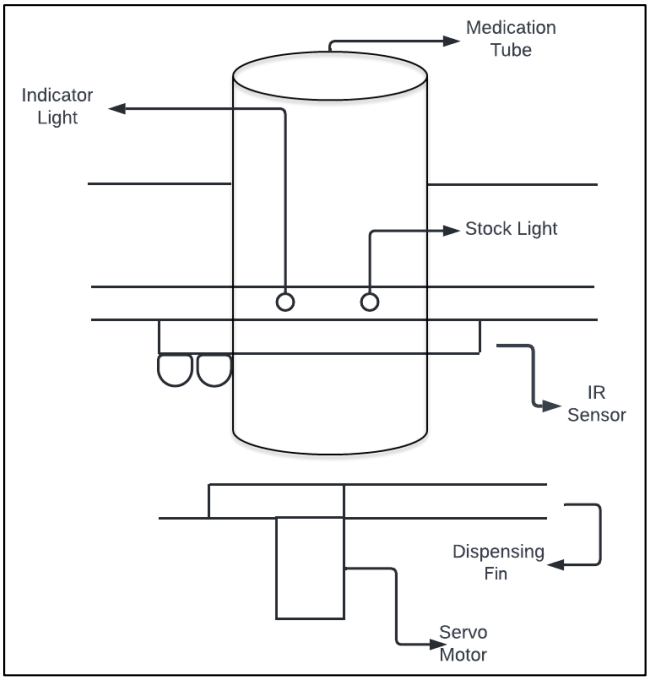


Figure A.4: Front View of the medicine dispensing system.

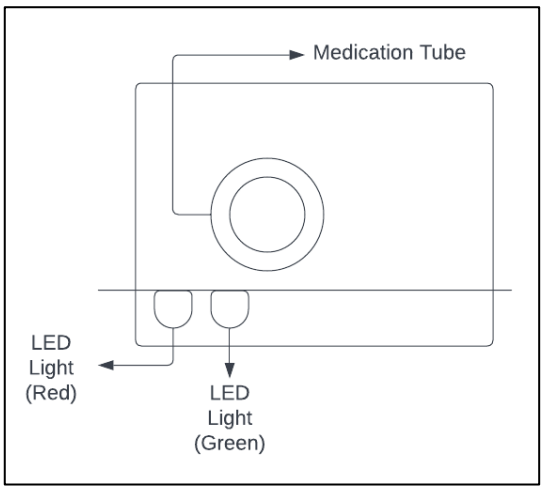


Figure A.5: Top view of the medicine dispensing system.

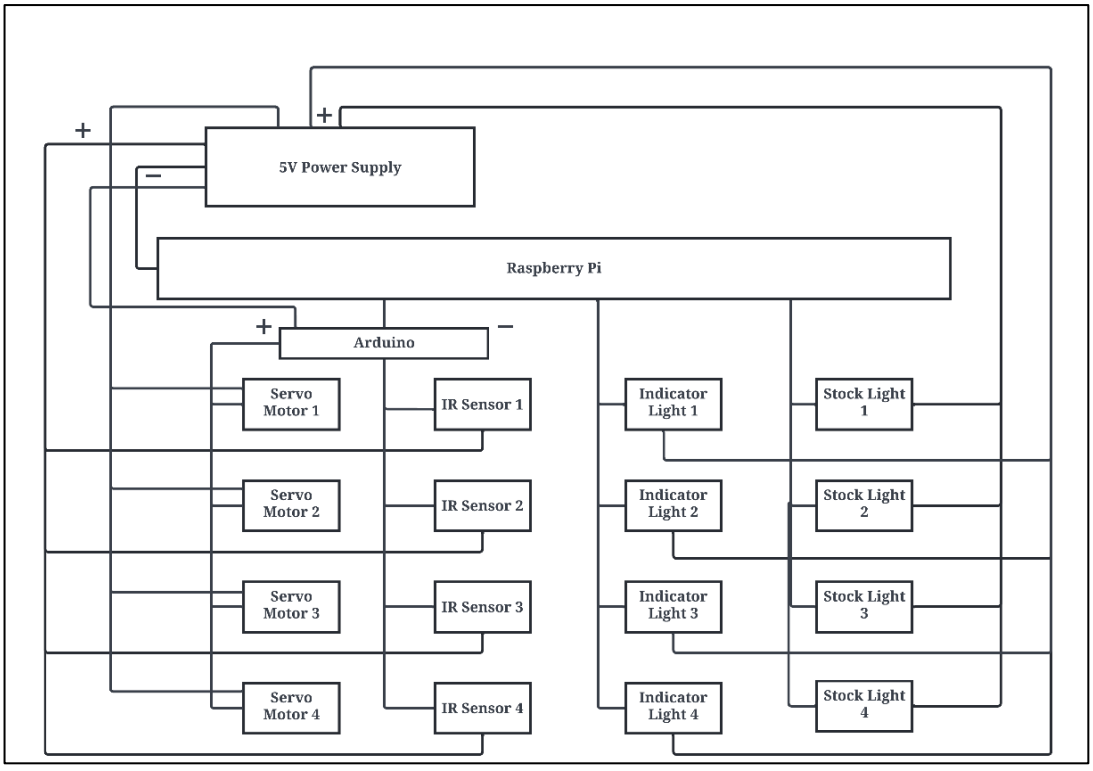


Figure A.6: Connection between the Power Supply, Raspberry Pi, Arduino, Servo Motors, IR Sensors and LED lights.

GITHUB LINK

GITHUB LINK

The below GitHub link contains the code of our project and report in a PDF format.

Link: https://github.com/Major-Project-22

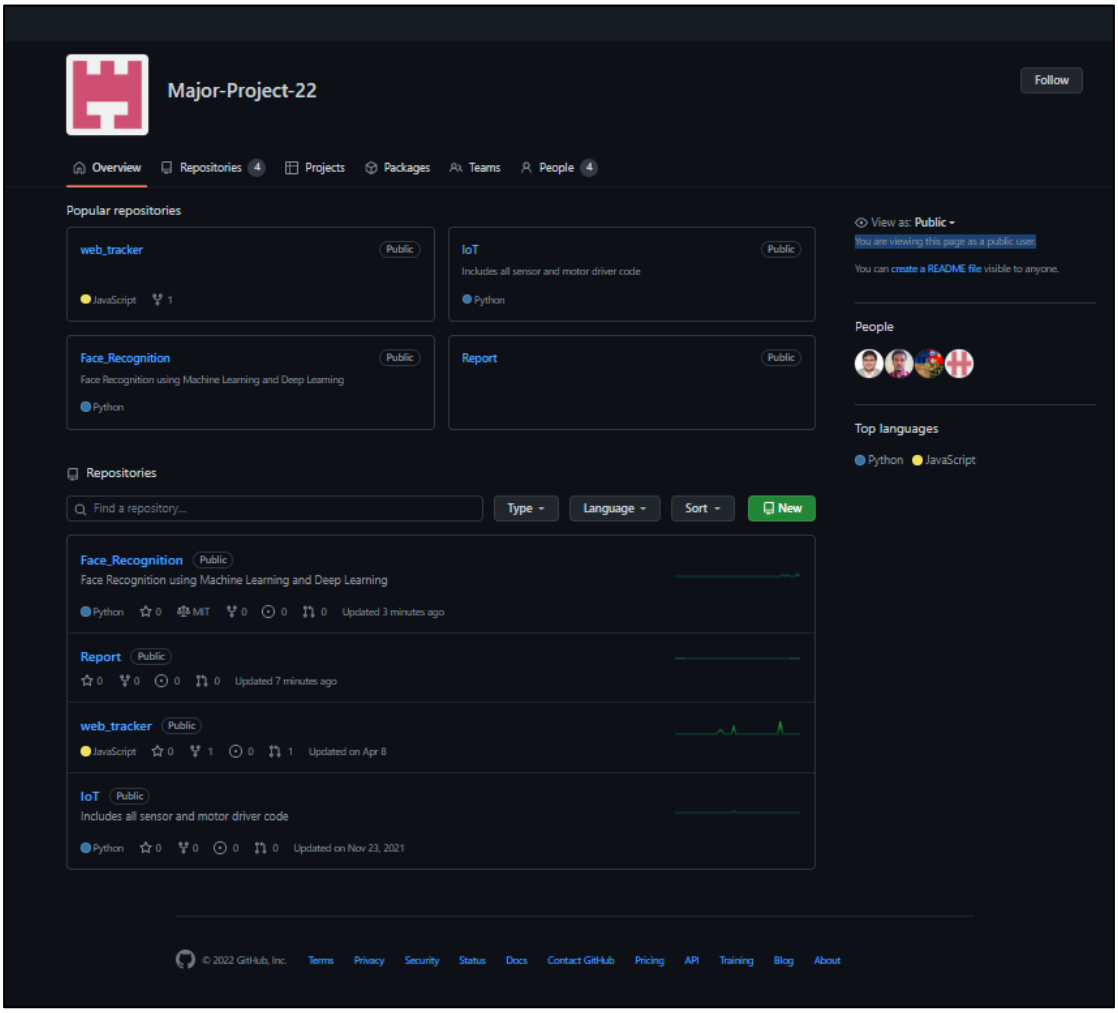
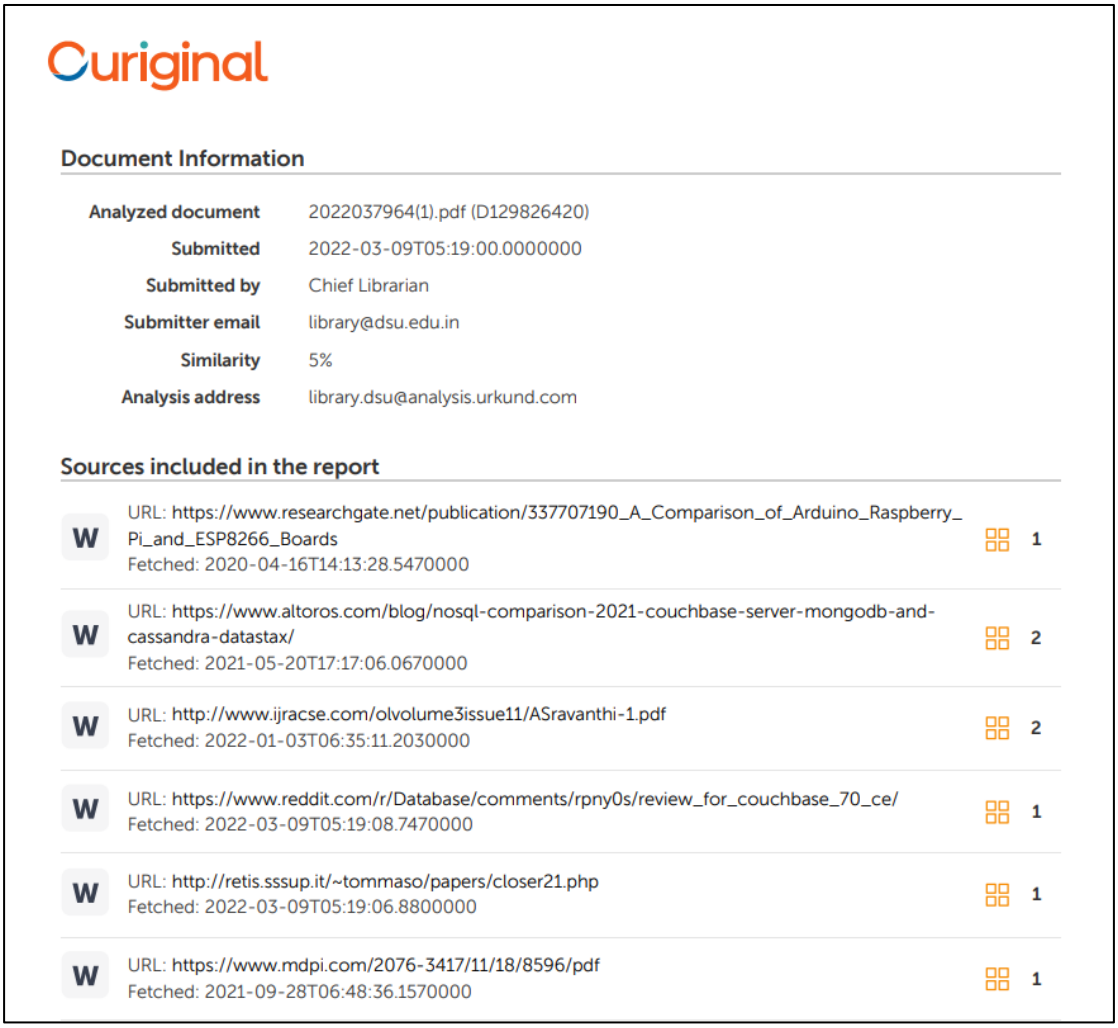


Figure: GitHub link containing the code and the report of our project.

PUBLISHED PAPER DETAILS IN IEEE ICDCECE 2022 CONFERENCE

PUBLISHED PAPER DETAILS IN IEEE ICDCECE 2022 CONFERENCE

Our paper titled "Analysis of Technologies used in Automatic Medicine Dispensing Systems" has been accepted at the IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE) 2022 Conference. The paper was presented on the 23rd of April, 2022.



Curiginal

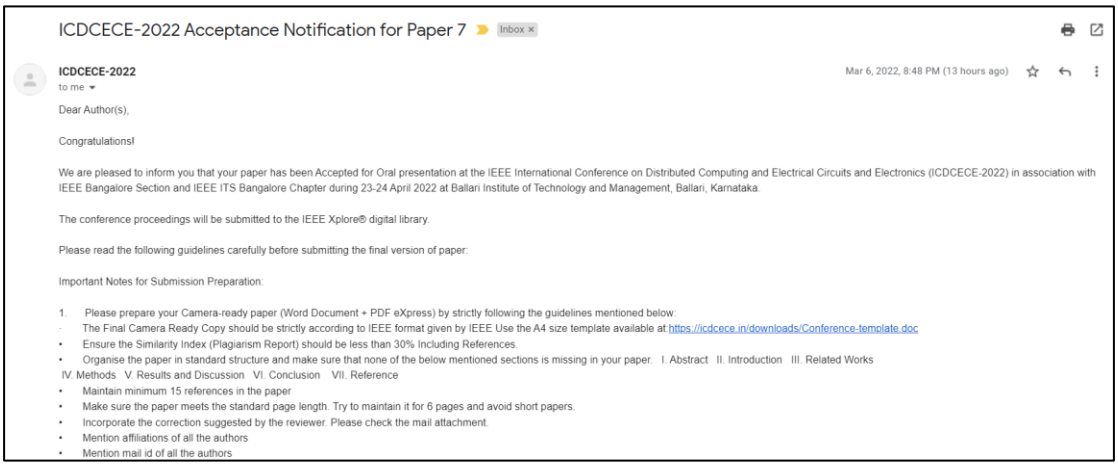
Document Information

Analyzed document	2022037964(1).pdf (D129826420)
Submitted	2022-03-09T05:19:00.0000000
Submitted by	Chief Librarian
Submitter email	library@dsu.edu.in
Similarity	5%
Analysis address	library.dsu@analysis.urkund.com

Sources included in the report

W	URL: https://www.researchgate.net/publication/337707190_A_Comparison_of_Arduino_Raspberry_Pi_and_ESP8266_Boards Fetched: 2020-04-16T14:13:28.5470000	1
W	URL: https://www.altoros.com/blog/nosql-comparison-2021-couchbase-server-mongodb-and-cassandra-datastax/ Fetched: 2021-05-20T17:17:06.0670000	2
W	URL: http://www.ijracse.com/olvolume3issue11/ASravanthi-1.pdf Fetched: 2022-01-03T06:35:11.2030000	2
W	URL: https://www.reddit.com/r/Database/comments/rpny0s/review_for_couchbase_70_ce/ Fetched: 2022-03-09T05:19:08.7470000	1
W	URL: http://retis.sssup.it/~tommaso/papers/closer21.php Fetched: 2022-03-09T05:19:06.8800000	1
W	URL: https://www.mdpi.com/2076-3417/11/18/8596/pdf Fetched: 2021-09-28T06:48:36.1570000	1

Figure: Plagiarism Check Report of the paper



ICDCECE-2022 Acceptance Notification for Paper 7 Inbox ×

ICDCECE-2022 to me — Mar 6, 2022, 8:48 PM (13 hours ago)

Dear Author(s),

Congratulations!

We are pleased to inform you that your paper has been Accepted for Oral presentation at the IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE-2022) in association with IEEE Bangalore Section and IEEE ITS Bangalore Chapter during 23-24 April 2022 at Ballari Institute of Technology and Management, Ballari, Karnataka.

The conference proceedings will be submitted to the IEEE Xplore® digital library.

Please read the following guidelines carefully before submitting the final version of paper:

Important Notes for Submission Preparation:

1. Please prepare your Camera-ready paper (Word Document + PDF eXpress) by strictly following the guidelines mentioned below:
   - The Final Camera Ready Copy should be strictly according to IEEE format given by IEEE Use the A4 size template available at https://icdcece.in/downloads/Conference-template.doc
   - Ensure the Similarity Index (Plagiarism Report) should be less than 30% Including References.
   - Organise the paper in standard structure and make sure that none of the below mentioned sections is missing in your paper. I. Abstract II. Introduction III. Related Works IV. Methods V. Results and Discussion VI. Conclusion VII. Reference
   - Maintain minimum 15 references in the paper
   - Make sure the paper meets the standard page length. Try to maintain it for 6 pages and avoid short papers.
   - Incorporate the correction suggested by the reviewer. Please check the mail attachment.
   - Mention affiliations of all the authors
   - Mention mail id of all the authors

Figure: Paper Acceptance Notification Email

IEEE International Conference on
Distributed Computing and Electrical Circuits and Electronics
ICDCECE- 2022

Organized by
Ballari Institute of Technology and Management, Ballari

PARTICIPANT

This Certificate is presented to Dr./Prof./Mr./Mrs.
Nobin Jaison Johnson

For His/Her Paper Titled
Analysis of Technologies used in Automatic Medicine Dispensing Systems

in the IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE-2022), organized by the Ballari Institute of Technology and Management, Ballari, India in association with IEEE Bangalore Section and IEEE Information Society on 23rd-24th April, 2022.

Dr. Abdul Lateef Haroon P S
Publication Chair
ICDCECE-22

Dr. Yadavalli Basavaraj
Principal
BITM

Mr. Prithviraj Y J
Trustee/Dy. Director
BITM

Figure: Paper Presentation Certificate – Nobin Jaison Johnson

Figure: Paper Presentation Certificate – Polaki Monika

Figure: Paper Presentation Certificate – Prajwal S

Figure: Paper Presentation Certificate – Thimmareddygari Jagadeeswar Reddy

Analysis of Technologies used in Automatic Medicine Dispensing Systems

Karishma Chavhan
Dept. of Computer Science
Dayananda Sagar University
Bangalore, India
karishma-cse@dsu.edu.in

Nobin Jaison Johnson
Dept. of Computer Science
Dayananda Sagar University
Bangalore, India
nobinjohnson3@gmail.com

Polaki Monika
Dept. of Computer Science
Dayananda Sagar University
Bangalore, India
polakimonika2000@gmail.com

Prajwal S
Dept. of Computer Science
Dayananda Sagar University
Bangalore, India
sathyaprajwal@gmail.com

Thimmareddygari Jagadeeswar Reddy
Dept. of Computer Science
Dayananda Sagar University
Bangalore, India
jagadeeswar.jd@gmail.com

***Abstract*—This paper attempts to compare the technologies used in Automatic Medicine Dispensing Systems using various aspects and benchmarks. Some of the modules involved in them are Person recognition using unique feature identification, the use of motors and detectors with IoT (Internet of Things), Database Management, and Web development.**

An automated Medication Dispensing System detects the patient using unique features such as face identification of id detection. The system then fetches the medication prescribed to the identified patient from the database. The system then uses servo motors to dispense the tablet from a tube containing the tablets. When the patient reaches out their hand, the IR sensor detects the presence of an object, and the medication is dispensed. The quantity of the medication is updated in the database and displayed on a webpage. The update in the quantity updates in the webpage in real-time. The system is user-friendly and efficient.

Keywords— Cloud Computing, Database Management System, Machine Learning, Internet of Things, Web Development.

I. INTRODUCTION

In the modern era, day-to-day activities that were once thought simplistic have been complicated. The medical industry in particular has been burdened by it. Simple tasks like dispensing medication require special priorities such as no contact dispensing. This paper attempts to give a detailed analysis and comparison between the latest technologies used in each domain to determine the best system to use in each required field.

A. Abbreviations

ML- Machine Learning

HTML – HyperText Markup Language

CSS - Cascading Style Sheets

JS - JavaScript

IoT – Internet of Things

JSON - JavaScript Object Notation

UI - User Interface

XHR - XMLHttpRequest

GPIO - General Purpose Input Output

CPU – Central Processing Unit

IR - InfraRed

LBP - Local Binary Pattern

PCA - Principal Component Analysis

KNN - K Nearest Neighbors

HRPSM-CNN - Hybrid Robust Point Set Matching Convolutional Neural Network

II. RELATED WORKS

A. Internet of Things

There are various microcontrollers and microprocessors available for use. Ex: Arduino, Raspberry Pi, Adafruit, etc. The above-mentioned devices are tailor-suited for different scenarios and use. There are more than 15 microcontrollers from Arduino and a large range of accessories or attachments for each microcontroller that performs a specific function such as a Bluetooth or Wi-Fi module. Each ranges from the difference in PINs, RAM, and ROM. The Arduino IDE features a simple version of C++ used to write the pin code [1]. The program written for one Arduino Microprocessor may not be compatible with another Arduino board. Multiple programs may not be executed at once unless the higher-end board is used. Hence using an Arduino for this system may not be feasible.

The Raspberry pi is a single board computer consisting of RAM, ROM, CPU, and GPU. It also contains a set of GPIO pins as seen in figure 1 making it useful in the IoT domain. Initially created for teaching and learning, the Raspberry Pi has grown in versatility and is being used commonly in every technical field including image processing, web development, cloud computing, etc.

The presence of a minimum of 1GB ram in all models and the smaller number of models available gives an advantage over most Arduino models which have memory in terms of kb. There are many models of raspberry pi available on the market. Previous iterations of this system have used the Raspberry Pi 2B which is limited in RAM. Hence

978-1-6654-8316-2/22/$31.00 ©2022 IEEE

multiprocessing was available to some extent. In this iteration, the raspberry Pi 4B 8GB RAM model will be used, which consists of a faster CPU and GPU as well as 8GB of RAM. Hence multiple complex programs may be run simultaneously.

The Raspberry Pi allows the programs to be written in any language and to run multiple operations at once which are memory or CPU-intensive operations [3]. This allows the Pi to perform GPIO operations as well as host the tracker webpage. Thus, from the information available from table 1, the Raspberry pi would be ideal for use as a microcontroller in the system as well as a server to host the tracking webpage.

TABLE 1. COMPARISON BETWEEN ARDUINO AND RASPBERRY [3]

Parameters	Arduino	Raspberry Pi
Developer	Arduino	Raspberry Pi Foundation
Type	Single Board micro-controller	Mini Computer
Operating system	None	Linux
CPU	Atmel, ARM, Intel	ARM Cortex
Clock Speed	16MHz	1.2GHz
Memory	32KB	1-4GB
Storage	1KB	MicroSDHC Slot
Power	USB, Battery, Power Supply	USB, Power Supply
Operating Voltage	5V	5V
I/O Connectivity	SPI 12C UART GPIO	SPI DSI UART SDIOCSI GPIO

The dispensing system requires a physical mechanism of pushing tablets to the receiving area. This type of mechanism usually consists of 2 modules, a sensor to detect the receiver and a motor to push the product. Several motors exist on the market for use. The DC motor is not ideal as it consumes a lot of power and the angle of rotation or the number of spin cycles cannot be mentioned. The power given must be estimated and the spin cycle can be determined using trial and error. AC cannot be used in this context as the system is small and the weight of the product to be pushed is in the number of grams. Stepper motors and servo motors are ideal for this system. Stepper and servo motors can move divisions within a cycle and return to the starting position [4], which is the desirable feature required for the system.

TABLE 2. COMPARISON OF ULTRASONIC AND IR SENSOR

Parameters	IR Sensor (SHARP GP2Y0A21YKOF)	Ultra-Sonic Sensor (HC SR-04)
Range	10cm-80cm	2cm-10m
Beam-width	75 Deg.	30 Deg.
Beam Pattern	Narrow (line)	Conical
Frequency	353 THz	40kHz

The sensing module has 2 options: an IR sensor and an Ultrasonic sensor. Both sensors are used for object detection but implemented in different ways. The IR sensor transmits IR rays and if they are reflected, the output is true. Thus, the IR sensor is only able to detect objects close by and not the distance of the object from the sensor. The Ultrasonic sensor works similarly but uses ultrasonic waves. The distance of the object can be calculated using the time taken for the wave to be received by the Transmitter [5]. Thus, the ultrasonic sensor has additional functionality, which is not required. This adds unrequired calculations to the system. Hence, from the information given in table 2, the IR sensor is the ideal sensing mechanism required for the system.

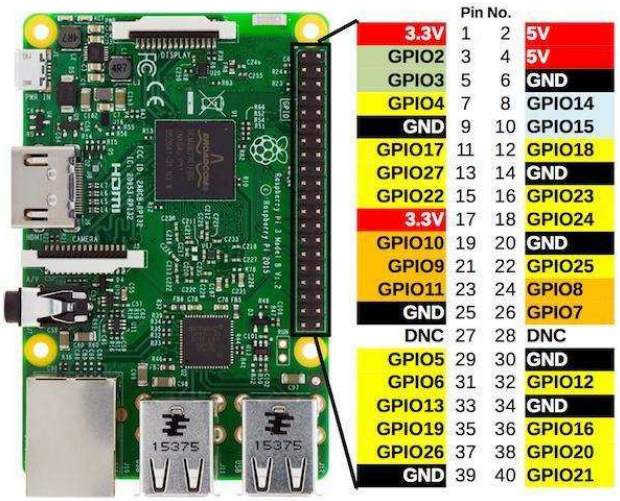


FIG. 1. GENERAL-PURPOSE INPUT-OUTPUT (GPIO) PINS.

B. Database Management Systems

With so many different types of database systems on the market today, deciding which type and database system to use has become a difficult task. This evaluation focuses on the performance of CRUD (CREATE, READ, UPDATE, DELETE) activities for various data volumes to show how these databases could be constructed and used in an application, as well as response time and complexity differences. [6]

The relational model is used by most popular database management systems (DBMS). The relational model, on the other hand, has many drawbacks that can be troublesome in some situations. The relational database model has a set of rules that must be followed. It means that a schema must be created ahead of time before data is loaded. The ACID features of RDBMS, on the other hand, hinder their ability to support significant scalability.

The requirement for high performance in cloud and scalable applications has accelerated the popularity of NoSQL (Not Only SQL (Structured Query Language) databases. Because NoSQL databases were designed to fit the widespread nature of the three-tier internet architecture, its persistent design and data format allow them to be easily partitioned on separate servers. [6]

Cassandra, being a member of the Column-Family, is relatively comparable to the classic relational model, which consists of columns and rows, when it comes to essential features. By examining basic qualities, it is possible to determine that common file formats, queries, transactions, locks, data storage, and operating systems have parallels.

MongoDB is a CP type system (consistency and partition tolerance), whereas Cassandra is a PA type system, according to the CAPs theorem (consistency and availability). Cassandra uses peer-to-peer replication, also known as Multi-master replication, rather than Master-Slave replication, as MongoDB does. In terms of usage and best application, MongoDB is preferable for Content Management Systems. [7]

Couchbase Server is a document-oriented and key-value distributed NoSQL database. It ensures fast speed by including an object-level cache, a SQL-like query language, asynchronous replication, ACID transactions, and data permanence. In reaction to workload, the database is set up to automatically scale resources such as CPU and RAM. If a user needs to run his database on a mobile device while also requiring multi-master replication, CouchDB is an obvious choice. Furthermore, if the database is rapidly growing, MongoDB is a better option than CouchDB. Even though both may be easily scaled over numerous nodes, MongoDB favors consistency over CouchDB. MongoDB is somewhat superior to CouchDB since it offers a SQL-like querying structure, although the former is easier to use. MongoDB also outperforms other databases when it comes to dynamic queries. In terms of database security, research is still ongoing, therefore it is difficult to predict. [8]

MongoDB outperformed MySQL in all core operations which is critical when supporting thousands of users at once. [9]. The MongoDB service is the main component, and it handles all of the essential database functions: it takes requests, applies changes to the storage unit, and manages and logs the database. MongoDB synchronously handles networking. Each incoming connection has its own server-side client worker thread that performs database operations and manages the session life cycle. MongoDB employs an optimistic variant of the Multi-version Concurrency Control technique, which enables concurrent write operations without locking [10].

C. Facial Recognition System

Facial recognition is an application that can detect, track, identify or verify human faces from an image or video captured using a camera. It is a difficult task to analyze the characteristics of the human face effectively and efficiently, which also requires a great deal of time. [11]

Facial recognition has 3 major important steps – detection of the face, extracting features from the facial image, and recognition of the face as given in figure 2.

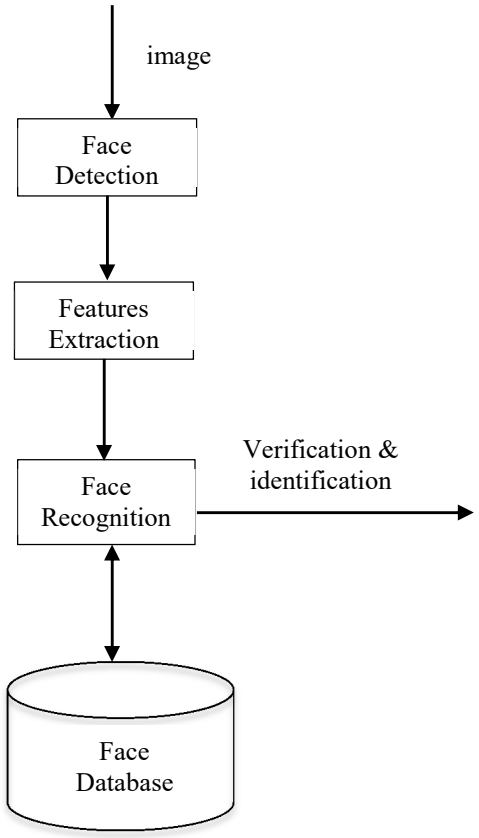


FIG. 2. FACE RECOGNITION STRUCTURE [12]

Various algorithms and image processing techniques exist such as contrast adjustment, bilateral filter, histogram equalizer, etc., to meet facial recognition needs. Image pre-processing techniques are used to process input images to have better features and characteristics that are compared with train datasets using the LBP algorithm. The input facial images are split into k^2 regions and the LBP code is deliberated for each pixel within the input facial image region; these LBP codes are subsequently compared with the center of the surrounding pixel. A binary 1 indicates that the surrounding pixel is above or equal to the central pixel, otherwise 0. A histogram is constructed for the majority of the regions and it represents a specific motif. These histograms are assembled to form a much larger histogram which is distinctive for each of them and is aggregated with facial images to recognize the faces. This technique enhances the LBP code and is precise for the facial recognition system. [11]

A variety of methods and techniques exist to create a facial recognition system. One of them is by making use of Principal Component Analysis (PCA) and K-Nearest Neighbors (KNN). The important steps/modules in a facial recognition system are capturing the input image i.e., the face, detecting the face and processing it; extracting the features, and finally accreditation. Principal Component is a modest method that makes use of eigenvector multivariate analysis. The number of principal components is always less than or equal to the number of original variables. The reason behind using KNN is that it is an uncomplicated classification algorithm. KNN classifies the mysterious data points into different classes among k closest examples. This technique is used to classify various face images. The

concept of Euclidean distance plays a major role; it is determined by calculating the distance between the testing and the training feature which results in a distance matrix. The performance of the technique is enumerated using the given below formula:

$$\text{Accuracy} = \frac{\text{truly detected facial image}}{\text{Gross facial images}} \quad (1)$$

Thus, PCA and KNN algorithms help develop an efficient facial recognition system. [13]

There are a few problems that arise in an effective and efficient facial recognition method that contains certain specific parameters. Although there are various algorithms like LBP, Directional Binary Code (DBC), Multi-Support Vector Machine (Multi-SVM), Convolutional Neural Networks (CNN), PCA, KNN, etc., face recognition is not achieving the desired result. A new algorithm known as Hybrid Robust Point Set Matching Convolutional Neural Network (HRPSM-CNN) efficiently recognizes faces under unconstrained circumstances. This method enhances the receiver operating characteristics in comparison to traditional algorithms. Images are pre-processed and then passed through the Viola-Jones algorithm in which features are extracted and classified into different classes. In the pooling layer of HRPSM-CNN, major important points are extracted and compared. This method is very efficient in recognizing the faces analyzing the parameters and then comparing them. This method shows greater recognition of 95% under various lighting conditions.

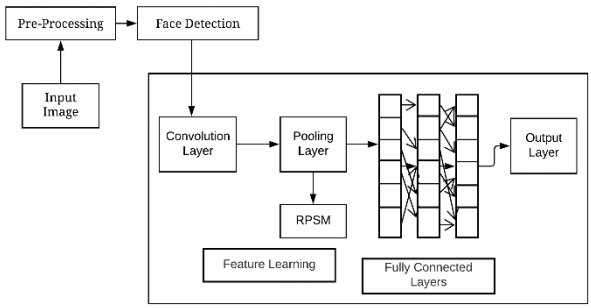


FIG. 3. BLOCK DIAGRAM OF HRPSM-CNN [14]

The above figure 3 throws light on the flow of the facial recognition process. The input image is processed and sent for face detection. In the pre-processing stage, the noise and other irrelevant features are removed from the image and then sent for face detection. Face detection is performed to predict whether the input image contains a human face or not. Under feature learning, convolution layer, pooling layer, and RPSM exist. In the convolution layer, filters are applied to blur and sharpen the images. The pooling layer plays an important role as it is used to minimize the dimensions of the feature maps. RPSM is used alongside CNN to enhance face recognition. The fully connected layers are the feed-forward neural networks. The output of the final pooling layer is the input to the fully connected layer. The output from the fully connected layer gives us the final output and accuracy. [14]

The LBP algorithm doesn't address the issues of occlusion and face mask recognition. In the technique that uses PCA and KNN, the value for k should be positive and the model does not work well with a large training data set. A facial recognition system needs a large training data set and this method of PCA and KNN is not efficient in producing accurate results. The HRPSM-CNN is the best-suited method for a facial recognition system as it makes use of a Convolutional neural network that needs no human interference. Also, by making use of the Viola-Jones algorithm, a greater accuracy rate can be achieved compared to the other two methods (i.e., LBP and PCA, KNN).

D. Web Development Application Tools Comparison

Researchers and Developers compare various web applications development tools to decide which technologies and tools to adopt in developing new web applications or frameworks to simplify the web application development process.

A. React

React can be used to develop custom front-end components and build complex user interfaces. Hence it is ideal for the implementation of a web tracker in this system.

B. Angular

Cross-platform applications with excellent speed and performance can be developed with Angular. The main disadvantage is its size, which might have a detrimental influence on web app performance.

Angular adapts a structural and modular approach and implements features like dependency injection, code generation, RESTful API, two-way data binding, component router, and others. [15]

C. Vue

VueJS is an open-source progressive JavaScript framework for building user interfaces. VueJS focuses on the view layer but not the other aspects of the application like Routing, HTTP requests. VueJS has a rich ecosystem of other powerful libraries that can be integrated. It's mainly used for building lightweight web apps such as single-page applications as well as user interfaces but lacks support for larger projects.

D. Express JS

ExpressJS is used for designing and building web applications quickly and easily. Express is lightweight and helps organize web applications on the server-side into a more organized MVC architecture. Single-page, multi-page, and hybrid web applications using ExpressJS.

E. Flask

Flask is a microframework web written in the python programming language; it is designed to make web

applications quickly and easily, with the ability to improve complex applications. Flask is used in top companies like Netflix, Reddit, Mozilla, Uber, Airbnb. The advantage of using flask is it is compatible with other python modules and allows python modules to interact with each other, combined with python's syntax simplicity. Hence powerful applications with multi-modular systems can be created. Python flask is one of the most versatile backends and is used widely for its simplicity and efficiency [16]

III. METHODS

To create the most efficient and versatile medicine dispensing system, the following design is proposed.

The Raspberry Pi is used as a microcontroller for motor and sensor control. It is also used as a server system for the tracking webpage. The IR sensor is used in combination with a servo motor for the dispensing mechanism. The below figure 4 emphasizes on the circuit diagram of the system.

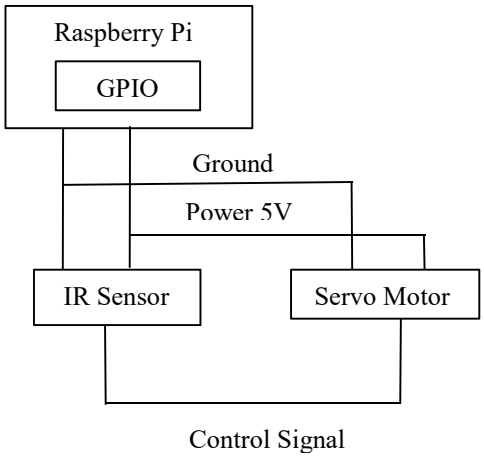


FIG 4. CIRCUIT DIAGRAM OF SYSTEM

MongoDB is used as the system database due to its document-oriented and no SQL nature. This makes the database efficient as no complex syntax is needed and database operations can be performed instantly and asynchronously. The database server can exist on the server machine or using cloud services. Local database servers have the advantage of speed due to time reduction in request and fetch operations.

The HRP-CNN algorithm is used for the facial recognition module. It is much more accurate and efficient when compared to other standard face recognition algorithms.

The MERN stack is used for the tracker webpage. Flask may be interchanged with Express as the back-end as there is a negligible difference in inefficiency. Using flask reduces the syntax complexity in performing database operations.

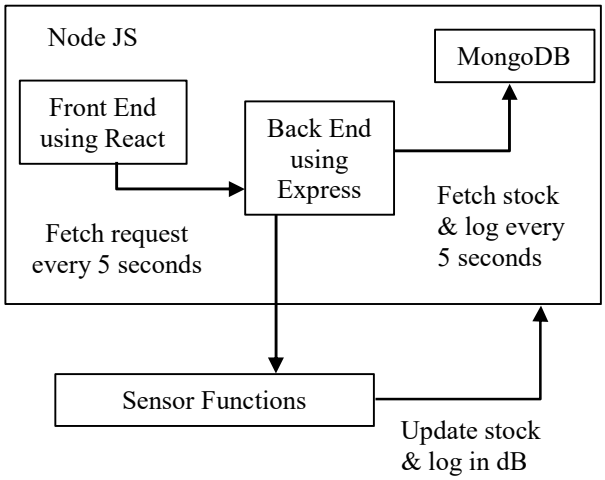


FIG 5. DATA FLOW DIAGRAM.

The above figure 5 depicts the data flow diagram of the system. Request is fetched every 5 seconds between the front end and the back end. After the sensor functions, the stock is updated in database. The stock and log is fetched every 5 seconds from the database.

IV. RESULTS AND DISCUSSION

The standard and upcoming techniques have been analyzed and the results are drawn. The comparison of different raspberry pi models is chosen according to benchmarking tests. The models underwent the whetstone, Dhrystone, double precision Lin pack, and sysbench benchmark tests. The performance is mentioned in the below figures.

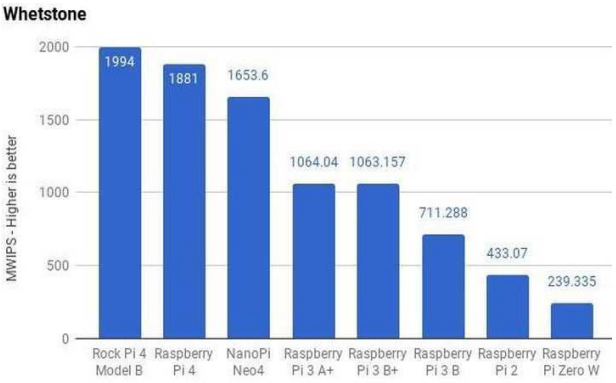


FIG 6. WHETSTONE BENCHMARK TEST.

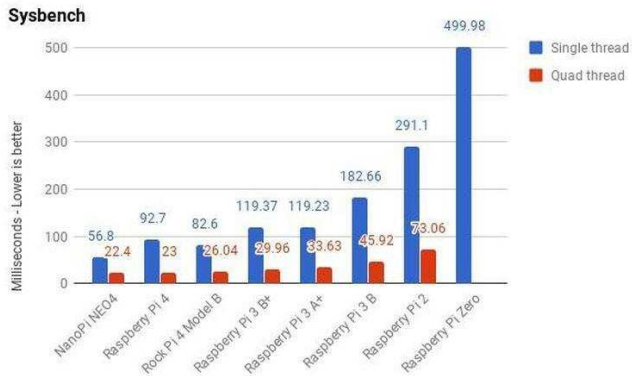


FIG 7. SYSBENCH BENCHMARK TEST.

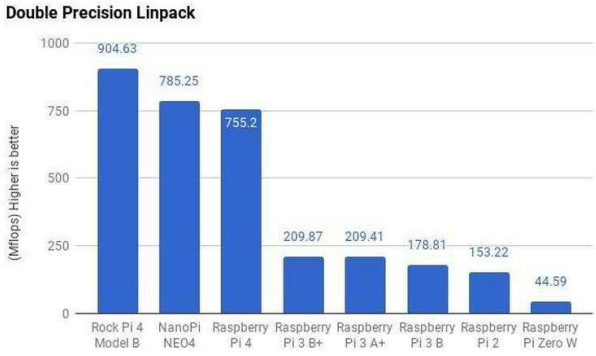


FIG 8. DOUBLE PRECISION LINPACK BENCHMARK TEST.

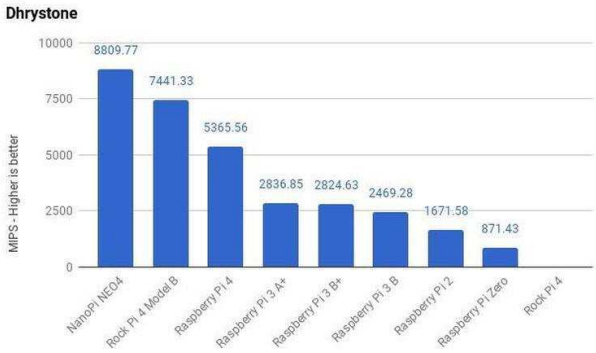


FIG 9. DHRYSTONE BENCHMARK TEST.

Among the face recognition models compared, the HRPSM-CNN algorithm has the highest precision and the comparison of the algorithms is given below.

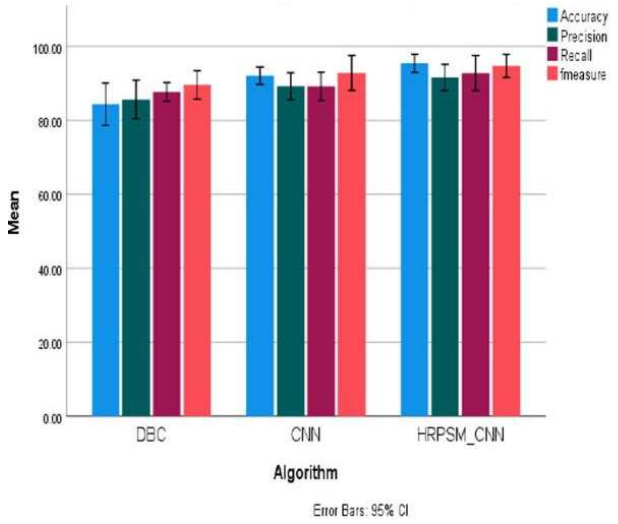


FIG 10. CLUSTERED BAR MEAN ACCURACY, MEAN OF PRECISION, MEAN OF RECALL, MEAN OF MEASURE [14]

The MongoDB Database provides higher throughput than other commonly available databases. The benefit of simple syntax makes it an ideal candidate for the system. The comparison of different databases is given below.

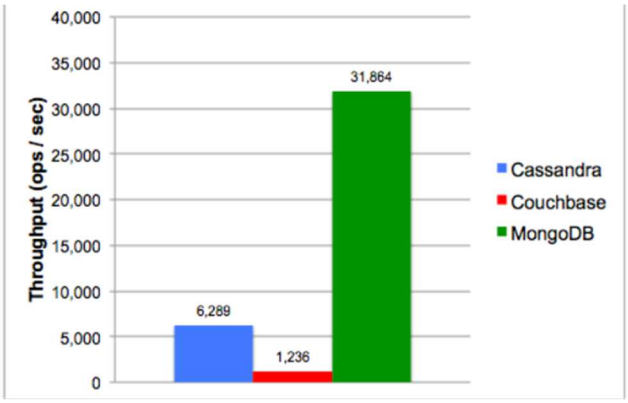


FIG 12. DATABASE *THROUGHPUT COMPARISON*

Web frameworks are plenty in number and have similar performance and throughput. It is dependent on the developer on the choice of framework. This makes MERN stack with flask instead of express efficient for the system.

The primary purpose of this research is to determine the performance of different technologies available in each domain required for this system. The Results revealed the performance of the available technologies through standardized benchmark and throughput tests. In addition, it also revealed the compatibility between different systems and interfacing among them.

In conclusion, the results are consistent with previous studies and give a clearer and more detailed analysis. With improved design and development in technologies and techniques, further analysis may provide a further optimized design making use of interconnected domains.

V. CONCLUSION

After the comparison of the latest tools available in these fields, it is concluded that the Raspberry pi will be used as the microcontroller due to its portable, volatile, and user-friendly nature. The IR sensor and servo motors will be used in conjunction due to their specific features which serve as requirements in this system. The best database for this particular system will be MongoDB as the webpage is a single route simplistic structure and MongoDB offers fast connection and operations with a NoSQL and documented oriented data structure. The HRPSM-CNN is the best-suited algorithm for facial recognition systems and it is much more efficient and sophisticated. The Backend for the tracker webpage will be created using Nodejs and express, due to its robust, minimal, and flexible set of features which is required for a single route webpage. In future iterations, an in-depth modular approach can be taken using a microservices architecture where each service is independent of others. ASP.NET Core can be used as the framework as it is the most efficient framework and can be paired with other services

such as cloud computing and the Internet of Things. This will also make the system portable, reliable, and available

VI. REFERENCES

[1] K. Ayush and R. Biswas, "Path follower & obstacle avoidance smart robot vehicle: Applications of arduino programming," *Int. J. Appl. Eng. Res.*, vol. 16, no. 6, p. 509, 2021

[2] J. F. Nusairat, "Raspberry Pi," in *Rust for the IoT*, Berkeley, CA: Apress, 2020, pp. 391–427.

[3] Ooko, Samson. (2019). A Comparison of Arduino, Raspberry Pi and ESP8266 Boards.

[4] H. Fattah, "Servo Motor," in *LTETM Cellular Narrowband Internet of Things (NB-IoT)*, CRC Press, 2021, pp. 319–324.

[5] X. Bi, "Infrared sensors and ultrasonic sensors," in *Environmental Perception Technology for Unmanned Systems*, Singapore: Springer Singapore, 2021, pp. 143–168.

[6] C. A. Győrödi, D. V. Dumşe-Burescu, D. R. Zmaranda, R. Ş. Győrödi, G. A. Gabor, and G. D. Pecherle, "Performance analysis of NoSQL and relational databases with CouchDB and MySQL for application's data storage," *Appl. Sci. (Basel)*, vol. 10, no. 23, p. 8524, 2020.

[7] A. Yudovin and C. Gutierrez, "NoSQL comparison 2021: Couchbase server, MongoDB, and Cassandra (DataStax)," *Altoros.com*, 30-Mar-2021. [Online]. Available: https://www.altoros.com/blog/nosql-comparison-2021-couchbase-server-mongodb-and-cassandra-datastax/. [Accessed: 09-Dec-2021].

[8] Kaur, R. & Sahiwal, J.K.. (2019). A review of comparison between NoSQL databases: MongoDB and couchDB. International Journal of Recent Technology and Engineering. 7. 892-898.

[9] A. Chauhan, "A Review on Various Aspects of MongoDb Databases," *Int. J. Eng. Res. Technol. (Ahmedabad)*, vol. 8, no. 5, 2019.

[10] R. Andreoli, T. Cucinotta, and D. Pedreschi, "RT-MongoDB: A NoSQL database with differentiated performance," in *Proceedings of the 11th International Conference on Cloud Computing and Services Science*, 2021.

[11] S. M. Bah and F. Ming, "An improved face recognition algorithm and its application in attendance management system," *Array*, vol. 5, no. 100014, p. 100014, 2020.

[12] Y. Kortli, M. Jridi, A. A. Falou, and M. Atri, "Face recognition systems: A survey," *Sensors (Basel)*, vol. 20, no. 2, p. 342, 2020

[13] I. Dinariyah and Alamsyah, "Accuracy enhancement in face recognition using 1D-PCA & 2D-PCA based on multilevel reverse-biorthogonal wavelet transform with KNN classifier," *J. Phys. Conf. Ser.*, vol. 1918, no. 4, p. 042144, 2021.

[14] M. Tamilselvi and S. Karthikeyan, "An ingenious face recognition system based on HRPSM_CNN under unrestrained environmental condition," *Alex. Eng. J.*, 2021.

[15] S. Ivanova and G. Georgiev, "Using modern web frameworks when developing an education application: a practical approach," in *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2019

[16] N. Chauhan, M. Singh, A. Verma, A. Parasher, and G. Budhiraja, "Implementation of database using python flask framework: College database management system," *Int. j. eng. comput. sci.*, vol. 8, no. 12, pp. 24894–24899, 2019.