Apeksha Hospital Donor Engagement System

2023-24-100

Team Members

Student Name	Student ID
Punchihewa S.N	IT20665166
Prabodha K.W.D.S	IT20665098
Bandara H.R.H.S	IT20662028
Wijesooriya P.L.P.G.D.S	IT20660352

Supervisor : Mrs.Lokesha Weerasinghe

Co-Supervisor : Ms.Chamali Pabasara

Introduction

Our project aims to revolutionize the donation process at Apeksha Hospital, ensuring high-quality donations and patient care.

Research Problem

The lack of a reliable and comprehensive platform for donations at Apeksha Hospital poses several challenges, including fraudulent activities and the absence of a specific and trustworthy platform for hair donations. These issues have a significant impact on the overall donation process, leading to inefficiencies, a lack of transparency, and a potential compromise in the quality and suitability of donated items for patient care.

Objectives

Main Objective

Improve communication, engagement, transparency, and security, while also optimizing donation campaigns and promoting high-quality donations.

Sub Objective

- Seamless Integration of Components
- User-Friendly Interface and Experience
- Scalability and Performance Optimization
- Continuous Monitoring and Improvement

Components

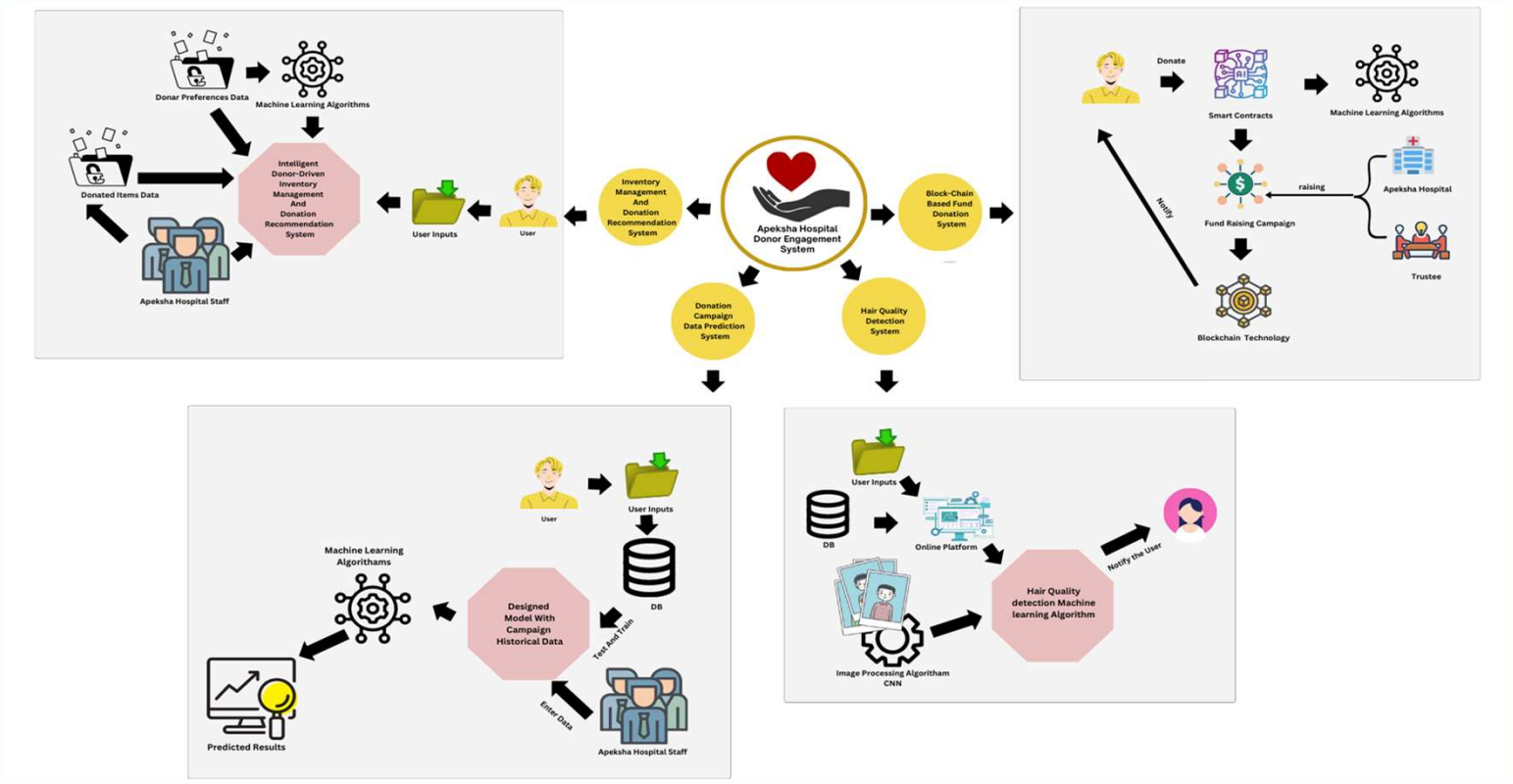
01 Intelligent Donor-Driven Inventory System for Essential Items

02 Critical Medication Priority Recommender System

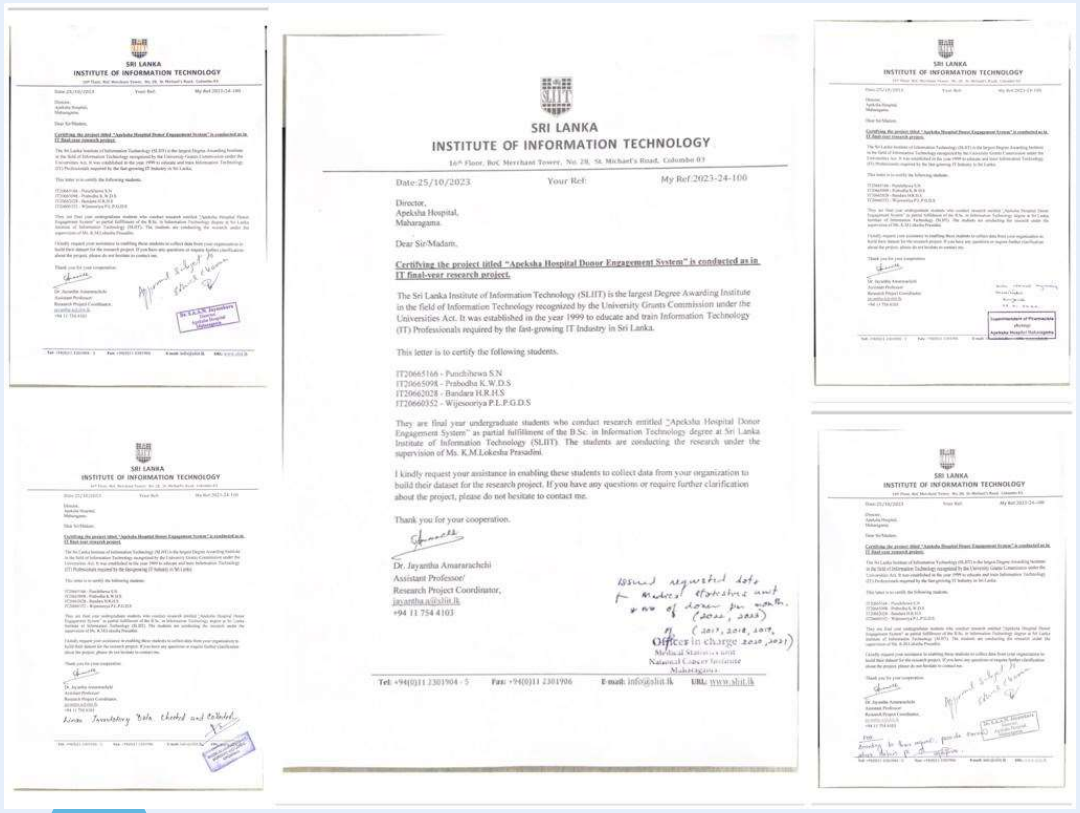
03 Predictive Analytics for Donation Campaign Success

04 Promoting quality hair donation for cancer patients

System Overview Diagram



Stakeholders Evidence



SRI LANKA
INSTITUTE OF INFORMATION TECHNOLOGY
16th Floor, BoC Merchant Tower, No. 28, St. Michael's Road, Colombo 03

Date:25/10/2023 Your Ref: My Ref:2023-24-100

Director,
Apeksha Hospital,
Maharagama.

Dear Sir/Madam,

Certifying the project titled "Apeksha Hospital Donor Engagement System" is conducted as in IT final-year research project.

The Sri Lanka Institute of Information Technology (SLIIT) is the largest Degree Awarding Institute in the field of Information Technology recognized by the University Grants Commission under the Universities Act. It was established in the year 1999 to educate and train Information Technology (IT) Professionals required by the fast-growing IT Industry in Sri Lanka.

This letter is to certify the following students.

IT20665166 - Punchihewa S.N
IT20665098 - Prabodha K.W.D.S
IT20662028 - Bandara H.R.H.S
IT20660352 - Wijesooriya P.L.P.G.D.S

They are final year undergraduate students who conduct research entitled "Apeksha Hospital Donor Engagement System" as partial fulfillment of the B.Sc. in Information Technology degree at Sri Lanka Institute of Information Technology (SLIIT). The students are conducting the research under the supervision of Ms. K.M.Lokesha Prasadini.

I kindly request your assistance in enabling these students to collect data from your organization to build their dataset for the research project. If you have any questions or require further clarification about the project, please do not hesitate to contact me.

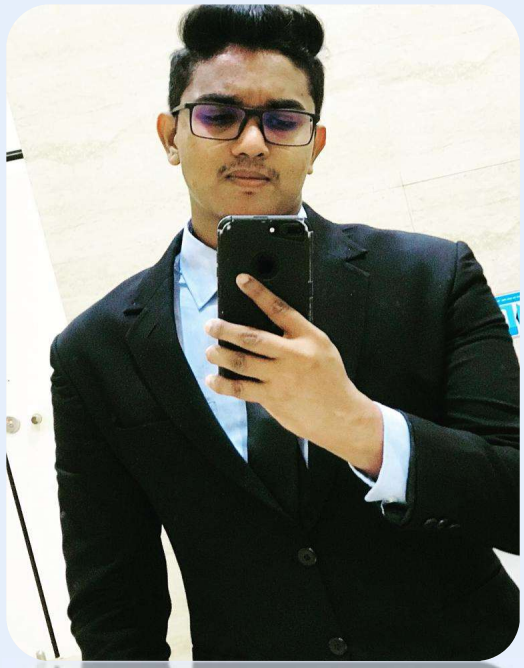
Thank you for your cooperation.

Dr. Jayantha Amararachchi
Assistant Professor/
Research Project Coordinator,
jayantha.a@slit.lk
+94 11 754 4103

Tel: +94(0)11 2301904 - 5 Fax: +94(0)11 2301906 E-mail: info@slit.lk URL: www.slit.lk

IT20665166
Punchihewa S.N

Bachelor of Science (Hons) in Information Technology
Specializing in Software Engineering

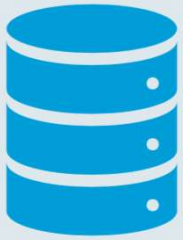


Introduction

Intelligent Donor-Driven Inventory System for Essential Items

IT20665166 | Punchihewa S.N. | 2023-24-100

Background



WHAT IS INTELLIGENT DONOR-DRIVEN INVENTORY SYSTEM?

USING MACHINE LEARNING TO UNDERSTAND ESSENTIAL ITEMS IN INVENTORY SYSTEM?

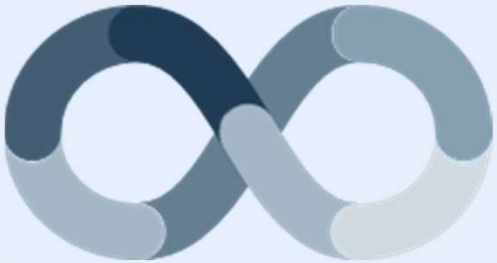
Problem

Intelligent donor-driven inventory management system. This solution utilizes machine learning algorithms to analyze Inventory behavior and historical Inventory usage patterns, recommending specific essential items required by the hospital. By providing an interactive interface for donors to select and track their impact, the system aims to enhance donor engagement and resource allocation efficiency.

Objective

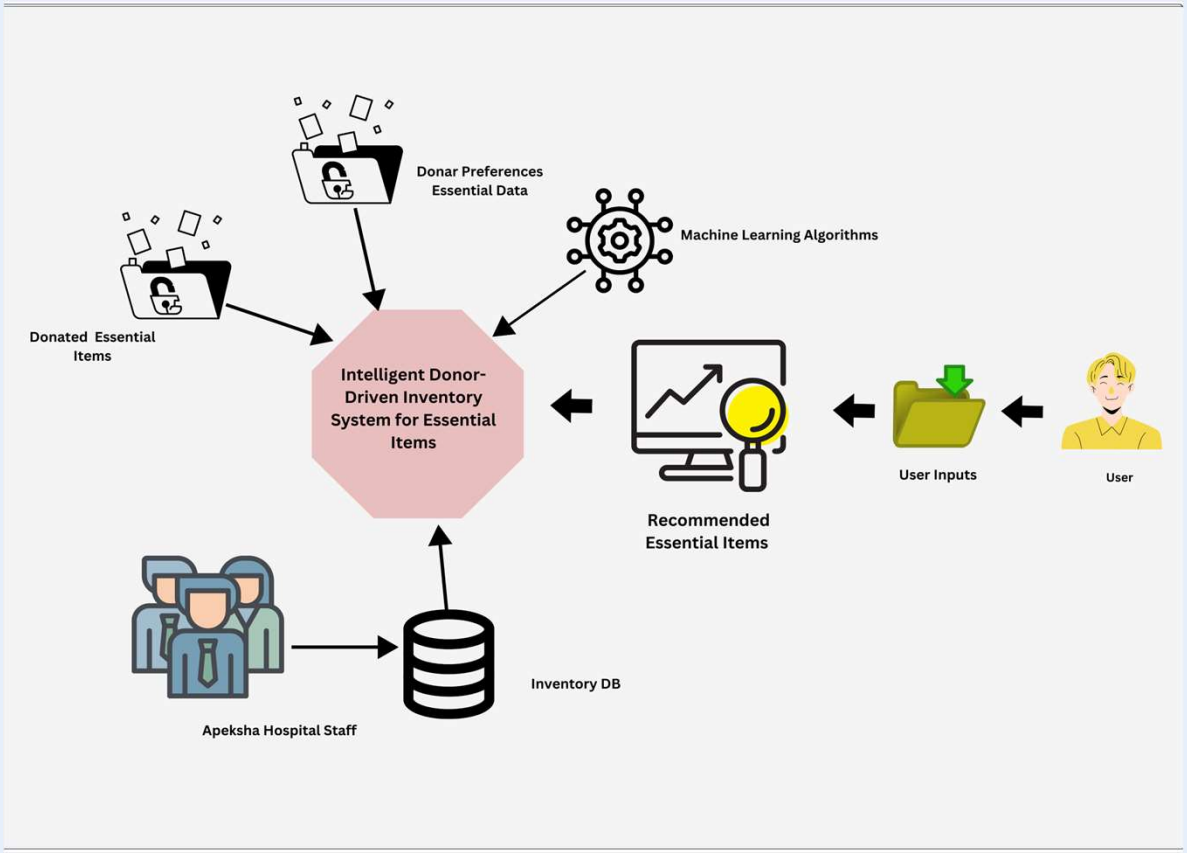
- **Develop predictive models utilizing historical Inventory data and relevant parameters to forecast inventory demand.**
- **Design a user-friendly dashboard enabling inventory managers to input variables and access real-time predictions.**
- **Establish a monitoring system to continuously update predictions as inventory variables evolve.**

Research Methodology



IT20665166 | Punchihewa S.N. | 2023-24-100

Component Diagram



Technologies and Techniques

Programming Language

- Python

Tools

- Jupyter Notebook
- Scikit Learn
- Anaconda Navigator

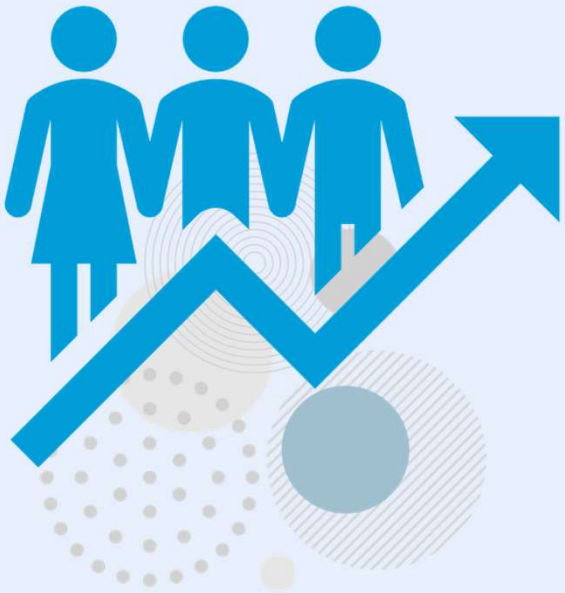
Version Controlling

- GitHub

Algorithm

- Decision Tree Regression

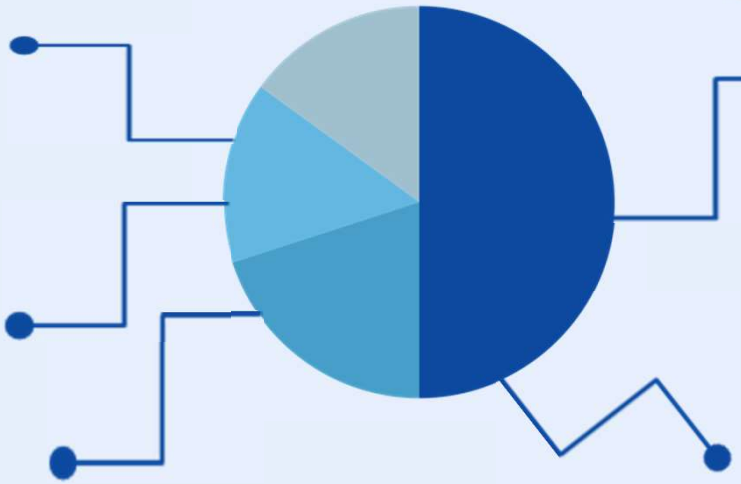
PROGRESS



Current Progress

Implement Machine Learning Backend using FastAPI

Deploy Machine Learning Model



Component integration 75% completed

Implement web application using MERN Stack Technology

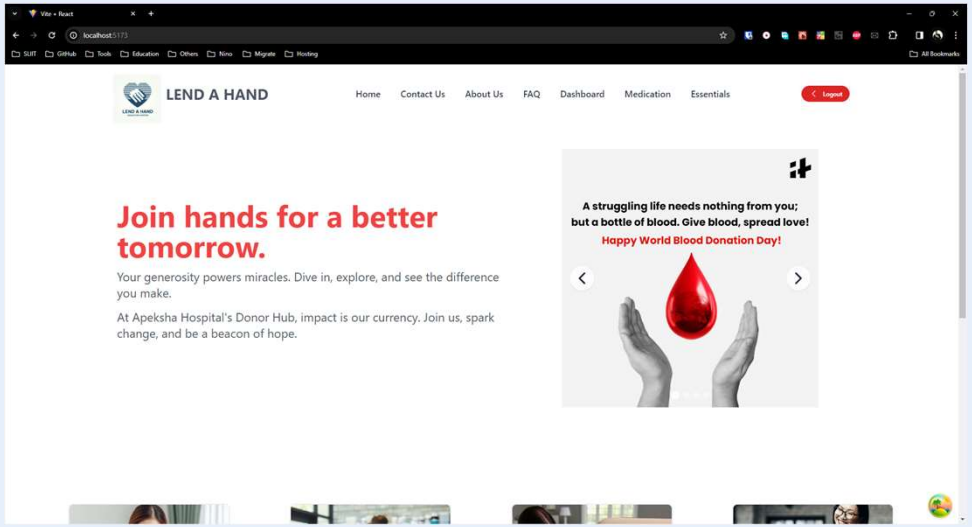
Identify the most suitable algorithm for Machine learning model train

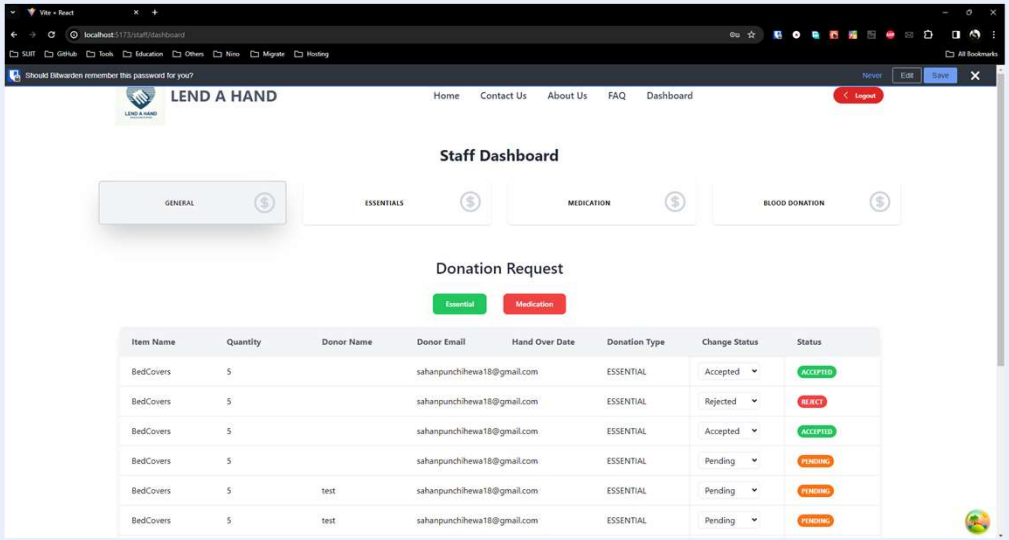
Next Expected Progress

- Enhance User Interface
- Complete component integration

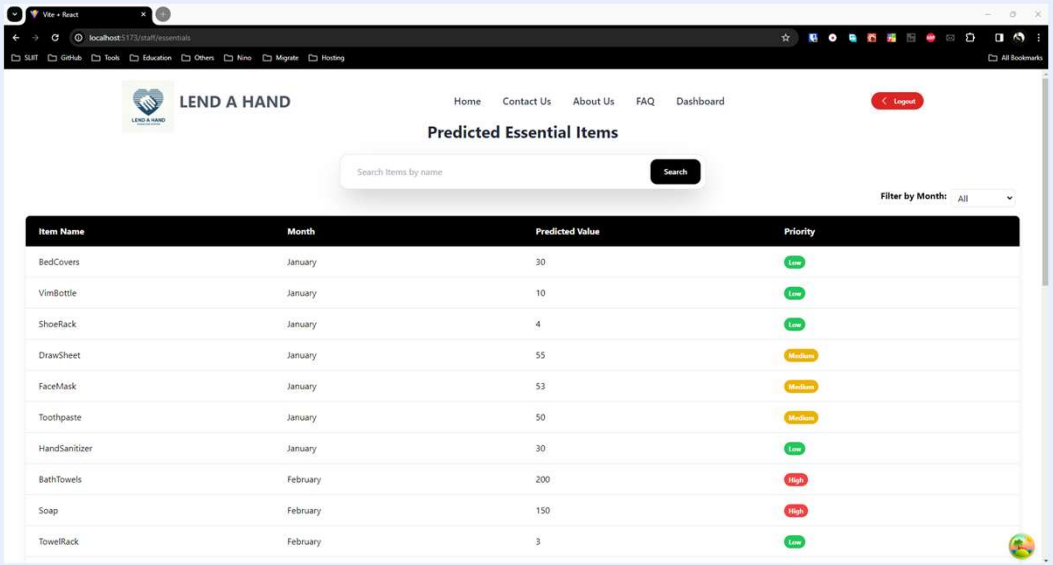
Project Evidence

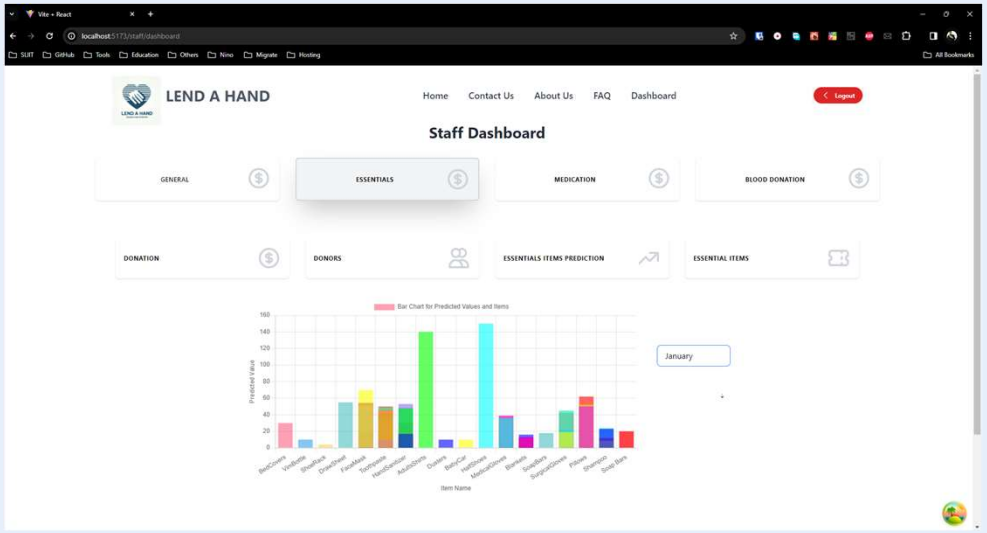
Project Interfaces





Machine Learning Implemention





Risk Mitigation

- **Data privacy and Security Risks**

Implement robust encryption methods and access controls to protect sensitive donor information. Regularly audit and update security measures to stay ahead of potential threats.

- **Donor trust and engagement**

Maintain transparent communication with donors about how their data is used and the benefits of the system. Highlight success stories and demonstrate the impact of donations to enhance donor trust and engagement.

Functional Requirements

- Create an interactive and user-friendly inventory management interface
- Essentials Items prediction
- Donation Utilization
- Intergration with hospital workflow
- Optimization

Non-Functional Requirements

IT20665098
Prabodha K.W.D.S

Bachelor of Science (Hons) in Information Technology
Specializing in Software Engineering

Critical Medication Priority Recommender System

Introduction

- **Research Problem**

We're addressing medication supply challenges at Apeksha Hospital with a 'Critical Medication Priority Recommender System.' Using machine learning and optimization, we aim to identify crucial medications, predict shortages, optimize procurement, and elevate patient care quality

- **Proposed Solution**

The system uses machine learning to predict shortages, identify critical medications, and optimize recommendations at Apeksha Hospital, improving patient care. It consists of a Shortage Prediction Model and a Critical Medication Identification Model.

Critical Medication Priority Recommender System

Introduction

- **Main Objective**

"Developing an intelligent system for Apeksha Hospital using machine learning to predict shortages, identify critical medications, and optimize their procurement, thus enhancing patient care."

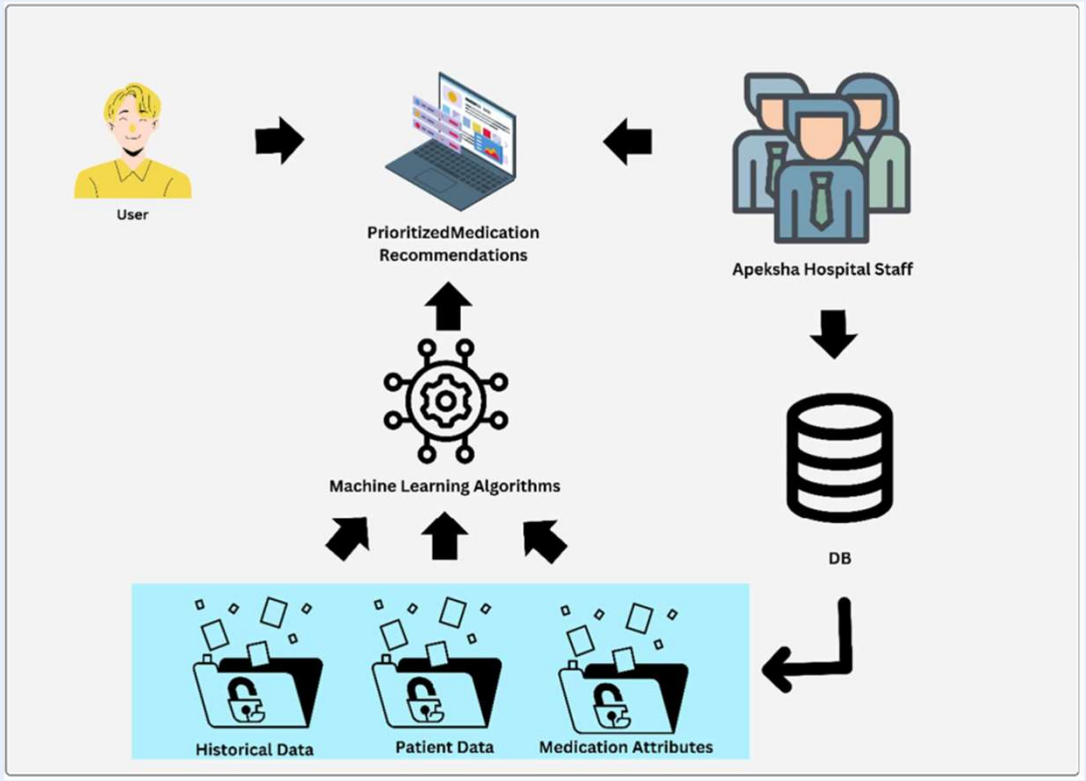
- **Specific Objectives**

1. Critical Medication Identification Model
2. Medication Shortage Prediction Model
3. Model Integration
4. User-Friendly Interface

Critical Medication Priority Recommender System

Methodology

- **Component Diagram**



Critical Medication Priority Recommender System

Methodology

- **Tools And Technologies**

Programming Language

- Python

Tools

- Jupyter Notebook
- Anaconda Navigator
- Scikit Learn
- Fast API
- Render

Version Controlling

- GitHub

Algorithms

- Random Forest Classification

Critical Medication Priority Recommender System

Methodology

Sub objective 01 - Critical Medication Identification Model

Data Collection

- Collected data from Apeksha Hospital regarding shortaged medications.

Data Pre-Processing

- Conversion of Categorical Variables to Numerical Format
- Date Column Conversion
- Date Feature Extraction

Implement Prediction Model

- Implement Machine Learning Model using "**Random Forest Classifier**" Algorithm

Critical Medication Priority Recommender System

Methodology

Sub objective 01 - Critical Medication Identification Model – Sample Codes

Random Forest Classifier **Accuracy- 86%**

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

# Load the dataset
data = pd.read_csv("hospital_dataset.csv")

# Drop rows with missing target values
data = data.dropna(subset=['Priority'])

# Separate features (X) and target (y)
X = data.drop(columns=['SR No', 'Item Name', 'Group', 'Priority'])
y = data['Priority']

# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a Random Forest classifier
model = RandomForestClassifier()
model.fit(X_train, y_train)

# Make predictions on the test set
predictions = model.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, predictions)

# Create a DataFrame with item names and predicted priority values
output_df = pd.DataFrame({'Item Name': data.loc[X_test.index, 'Item Name'], 'Predicted Priority': predictions})

# Display the output and accuracy
print("Predicted values:")
print(output_df)
print("\nAccuracy:", accuracy)

print("Train set shape:", X_train.shape)
print("Test set shape:", X_test.shape)


```

Critical Medication Priority Recommender System

Methodology

Sub objective 01 - Critical Medication Identification Model

Random Forest Classifier

```
Accuracy: 0.8604651162790697
Train set shape: (171, 3)
Test set shape: (43, 3)
```

Critical Medication Priority Recommender System

Methodology

Sub objective 02 - Medication Shortage Prediction Model

Data Collection

- Collected data from Apeksha Hospital.

Data Pre-Processing

- Conversion of Categorical Variables to Numerical format
- Date Column Conversion
- Date Feature Extraction
- Applied feature scaling using the StandardScaler from scikit-learn

Implement Prediction Model

- Implement Machine Learning Model using "**Logistic Regression** " Algorithm
- Implement Machine Learning Model using “**Random Forest**" Algorithm

Critical Medication Priority Recommender System

Methodology

Sub objective 02 - Medication Shortage Prediction Model – Sample Code

Logistic Regression Algorithm **Accuracy- 78%**

```python
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LogisticRegression
from sklearn.impute import SimpleImputer
from sklearn.metrics import accuracy_score
import joblib

# Load the dataset containing all necessary information
df = pd.read_csv("medication_shortage.csv")

# Convert date columns to datetime
df['shelf_life'] = pd.to_datetime(df['shelf_life'])
df['Admin_Timestamp'] = pd.to_datetime(df['Admin_Timestamp'])

# Extract date features
df['Shelf_Life_Day'] = df['shelf_life'].dt.day
df['Shelf_Life_Month'] = df['shelf_life'].dt.month
df['Shelf_Life_Year'] = df['shelf_life'].dt.year
df['Admin_Day'] = df['Admin_Timestamp'].dt.day
df['Admin_Month'] = df['Admin_Timestamp'].dt.month
df['Admin_Year'] = df['Admin_Timestamp'].dt.year

# Drop unnecessary columns
df.drop(['shelf_life', 'Admin_Timestamp'], axis=1, inplace=True)

# Split features and target variable
X = df.drop(columns=['shortaged', 'Name','ID'])  # Exclude non-numeric columns
y = df['shortaged']

# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=21)

# Define column transformer to handle numeric and categorical features
numeric_features = X.select_dtypes(include=['int64', 'float64']).columns
numeric_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='median')),
```

Random Forest Classifier **Accuracy- 91%**

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestClassifier  # Change
from sklearn.impute import SimpleImputer
from sklearn.metrics import accuracy_score

# Load the dataset containing all necessary information
df = pd.read_csv("medication_shortage.csv")

# Convert date columns to datetime
df['shelf_life'] = pd.to_datetime(df['shelf_life'])
df['Admin_Timestamp'] = pd.to_datetime(df['Admin_Timestamp'])

# Extract date features
df['Shelf_Life_Day'] = df['shelf_life'].dt.day
df['Shelf_Life_Month'] = df['shelf_life'].dt.month
df['Shelf_Life_Year'] = df['shelf_life'].dt.year
df['Admin_Day'] = df['Admin_Timestamp'].dt.day
df['Admin_Month'] = df['Admin_Timestamp'].dt.month
df['Admin_Year'] = df['Admin_Timestamp'].dt.year

# Drop unnecessary columns
df.drop(['shelf_life', 'Admin_Timestamp'], axis=1, inplace=True)

# Split features and target variable
X = df.drop(columns=['shortaged', 'Name','ID'])  # Exclude non-numeric columns
y = df['shortaged']

# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Define column transformer to handle numeric and categorical features
numeric_features = X.select_dtypes(include=['int64', 'float64']).columns
numeric_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='median')),
    ('scaler', StandardScaler())])

categorical_features = X.select_dtypes(include=['object']).columns
```

Critical Medication Priority Recommender System

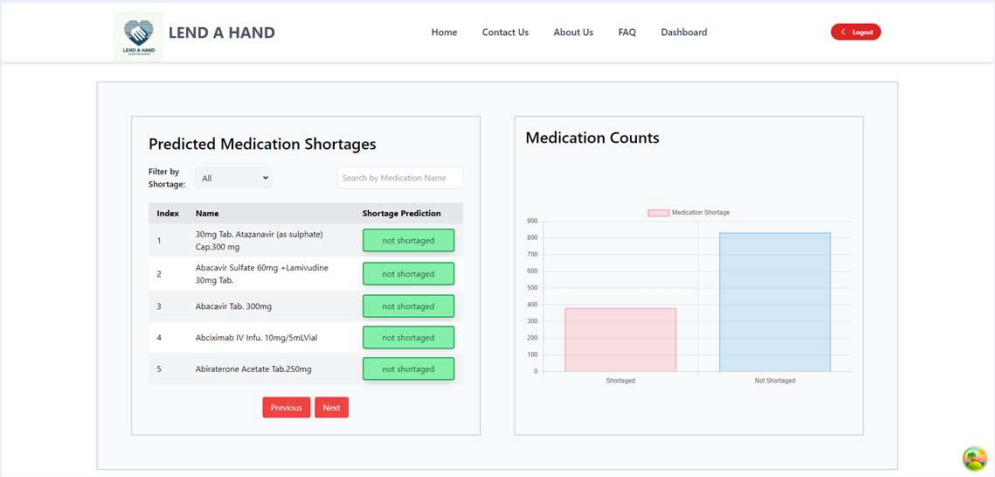
Methodology

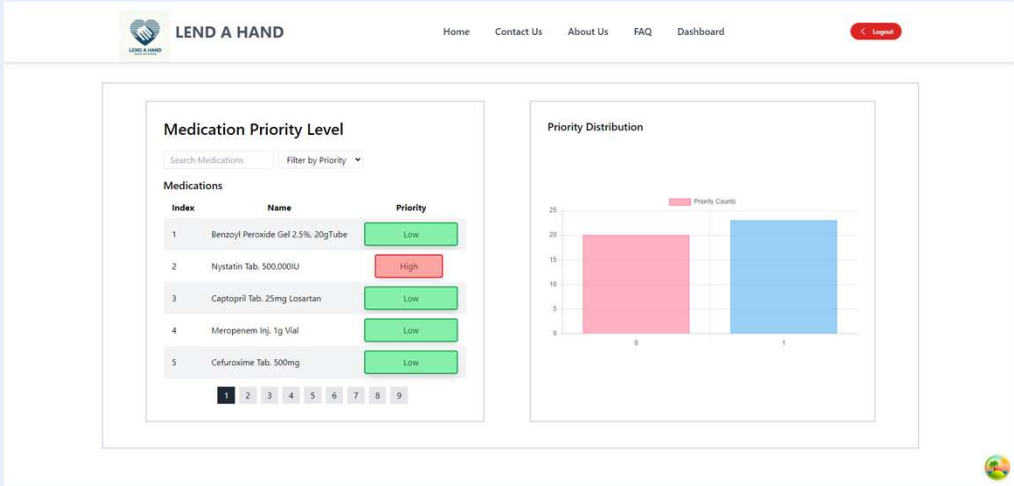
Sub objective 02 - Medication Shortage Prediction Model

Random Forest Classifier

```
      Predicted_Shortage_RF
0                         1
1                         1
2                         1
3                         1
4                         1
...                     ...
1207                      0
1208                      0
1209                      1
1210                      1
1211                      0

[1212 rows x 3 columns]
Random Forest Model Accuracy: 0.9125412541254125
```

Implemented Medication Dashboard

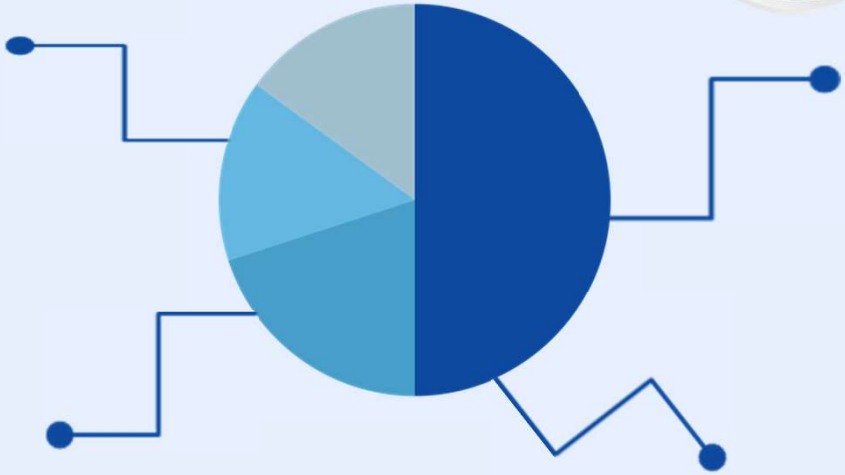




Critical Medication Priority Recommender System

Current Progress

Implemented Shortage Prediction model



Deploy Machine Learning Models

Implemented Critical medication Identification model

Model Integration and User Interface Implementation

Next Expected Progress

- Enhance User Interfaces and Complete UI implementation.
- Complete component integration

IT20662028
Bandara H.R.H.S

Bachelor of Science (Hons) in Information Technology
Specializing in Software Engineering

Introduction

Blood Donation Campaign Organizing Challenges

- There is High Demand for blood in Apeksha hospital.
- Organize donation Campaigns without Prediction.
- Food Wastage and Crowd Management Issues.
- Waste campaign marketing cost and another cost.

Research Problem & Solution

- Problem:

Apeksha Hospital's blood donation campaigns face challenges due to low donor participation and resource inefficiencies.

- Solution:

Using predictive analytics and machine learning, Apeksha Hospital can boost campaign effectiveness by predicting success factors and optimizing resource allocation.

Specific and Sub Objective

- **Main Objective**

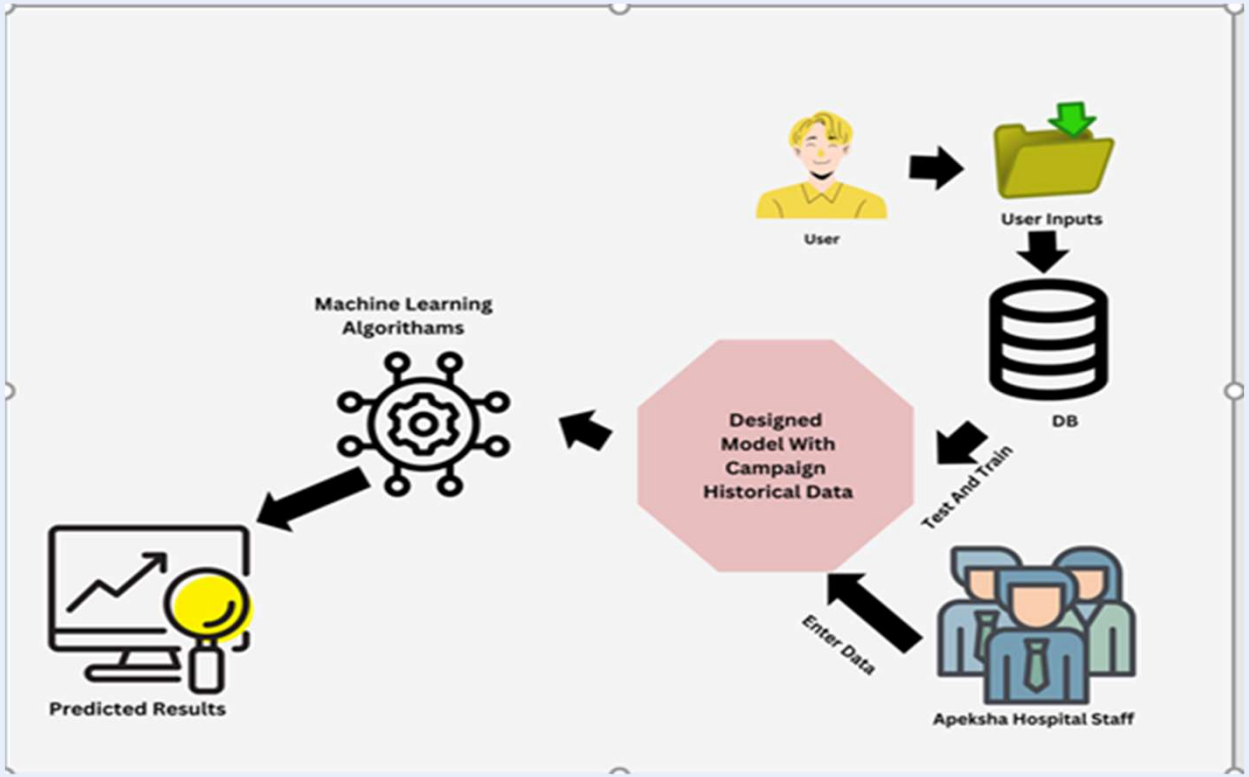
Implement machine learning to enhance the efficiency of blood donation campaigns.

- **Specific Objectives**

1.Optimization Algorithms Integration
2.Predict People amount
3.User-Friendly Interface

Methodology

- **Component Diagram**



Methodology

- **Tools And Technologies**

Programming Language

- ❖ **Python**
- ❖ **Java Script**

Tools

- ❖ **Jupyter Notebook**
- ❖ **Anaconda Navigator**
- ❖ **Scikit Learn**
- ❖ **Joblib**

Version Controlling

- ❖ **GitHub**

Algorithm

- ❖ `RandomForestRegressor`

Methodology
Evidence of Completion

Data collection

Data pre-processing

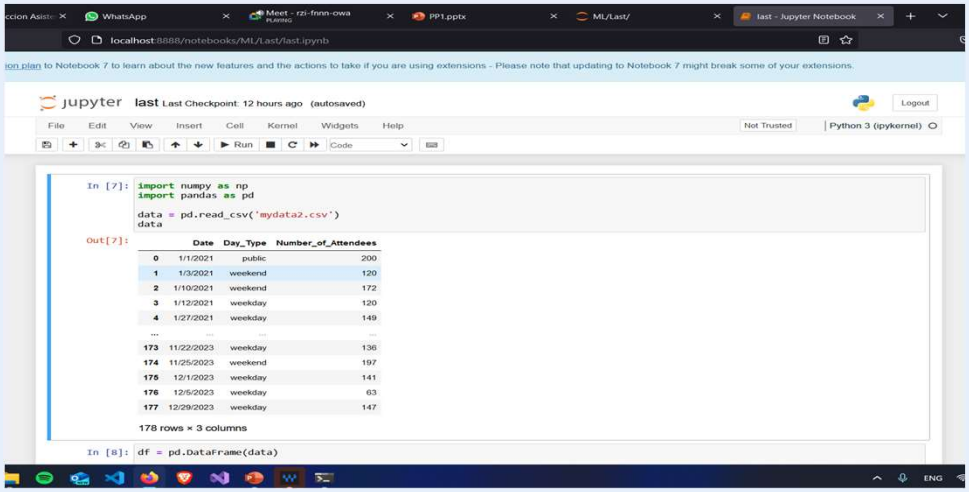
Used Random Forest Regressor Algorithm for training models

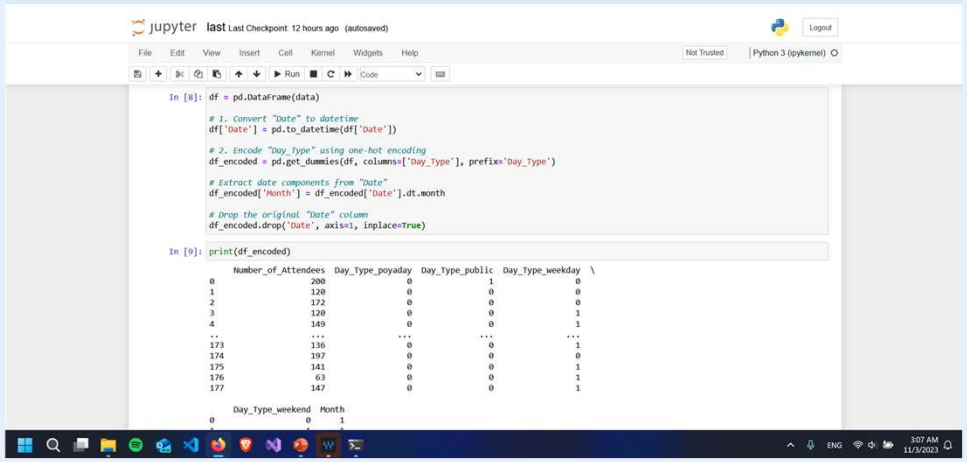
Data Visualization

Host the Current model in the flask server and Render

Displayed the prediction in web application

Data Collection and Pre-processing

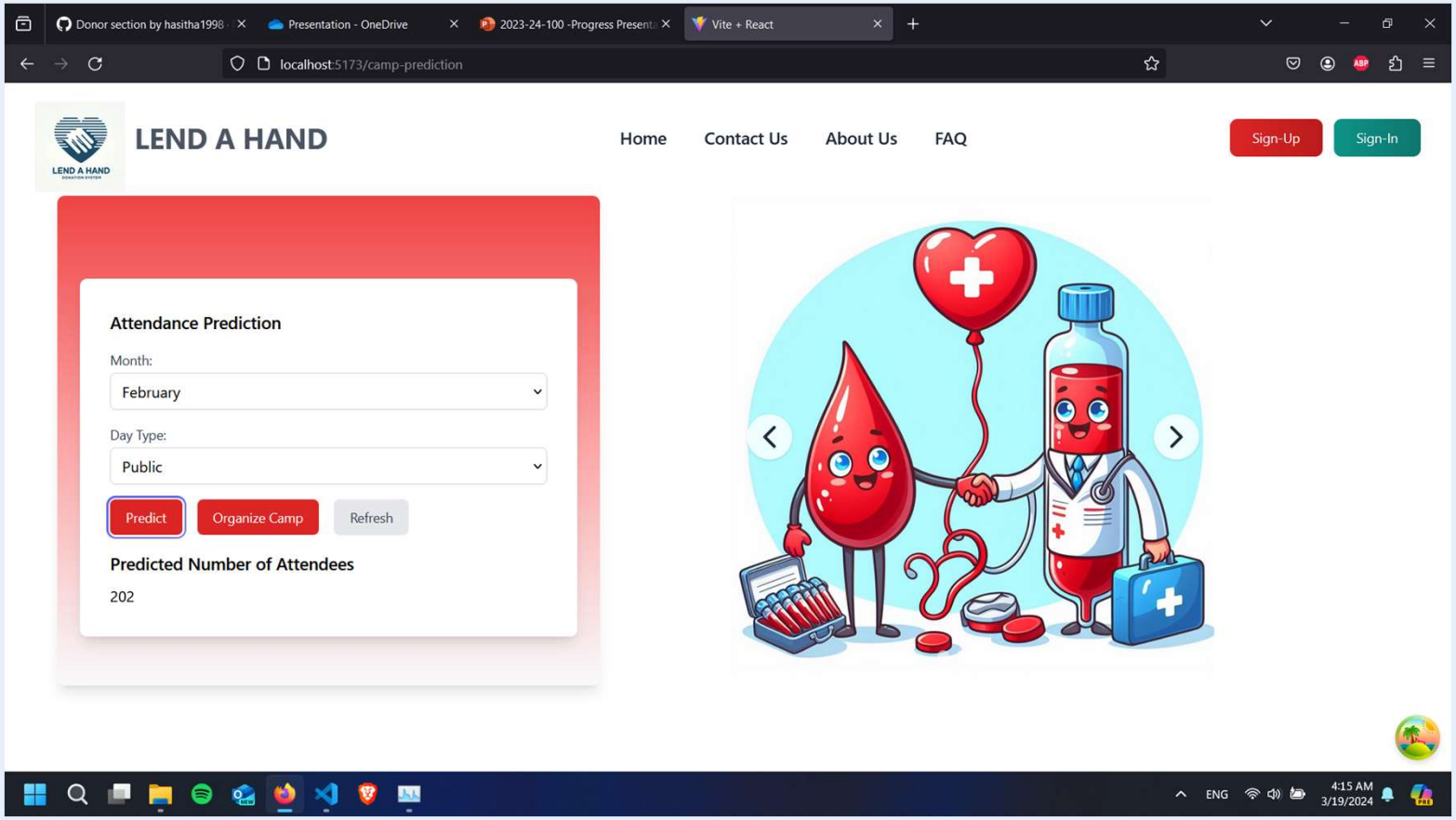




Random Forest Regressor Accuracy

```
Enter the month (1-12): 2
Is it a Poyaday? (0 or 1): 1
Is it a public holiday? (0 or 1): 0
Is it a weekday? (0 or 1): 0
Is it a weekend? (0 or 1): 1
Predicted Number of Attendees: 342.16
R-squared (R²) on the entire dataset: 0.9292643387482032
```

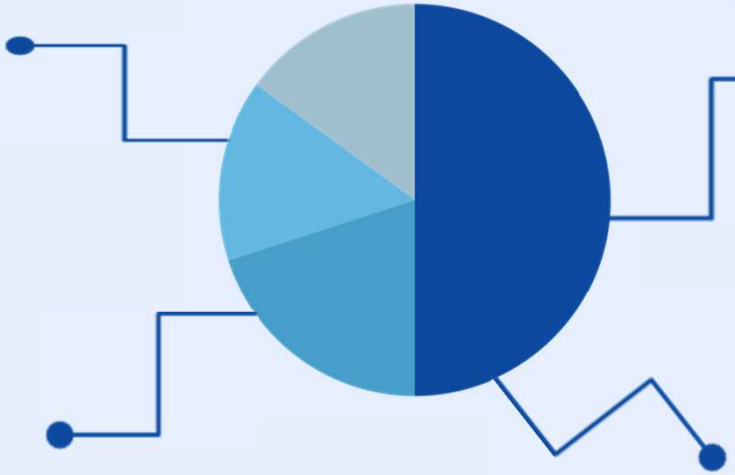
Display the results in Web UI



Critical Medication Priority Recommender System

Current Progress

Implemented people amount Prediction model

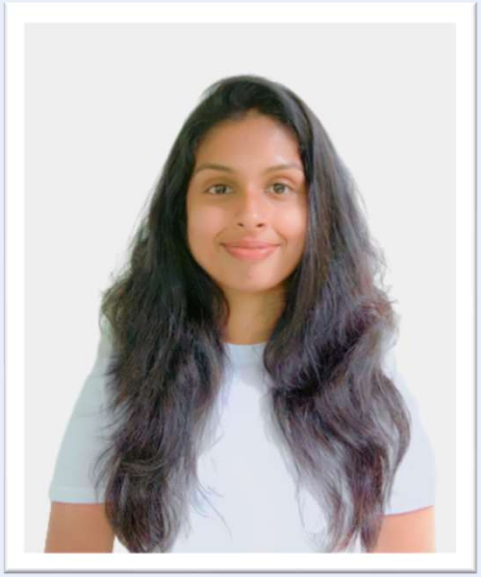


Deploy Machine Learning Models

Implemented blood camp organizers and donation camp

Model Integration and User Interface Implementation

IT20660352
Wijesooriya P.L.P.G.D.S

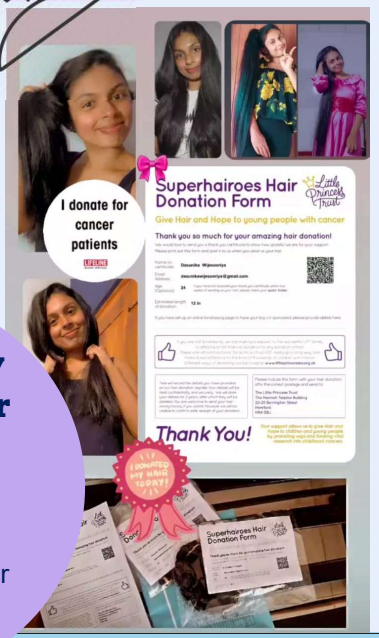


Bachelor of Science (Hons) in Information Technology
Specializing in Software Engineering

Promoting Quality Hair Donation for Cancer Patients

Introduction
Background

Apeksha hospital-based hair donation process currently has no specific method. Applying to build the system to obtain hair from hair donors according to the standards recommended by the hospital and to eliminate the distance between the hospital and the donor. Using CNN Architecture in image processing.



Research
Problem

How to determine whether hair donated by hair donors meets the standards and qualities of hair recommended by the Apeksha hospital?

I donated my hair to cancer patients.

But,

Apeksha Hospital could not donate hair due to the informal procedure there.

Specific and Sub Objective

Specific Objective

Implementation of Deep Learning Algorithm (CNN) with online platform to examine key quality factors of hair. – main objective

- **Model Implemented to Hair Identification- 1.1**
- **Model Implemented to identify Color of Hair -1.2**
- **Model Implemented to identify bleached of Hair – 1.3**
- **Model Implemented to identify Dryness of Hair – 1.4**
- **Model Implemented to identify lice & dandruff of Hair -1.5**

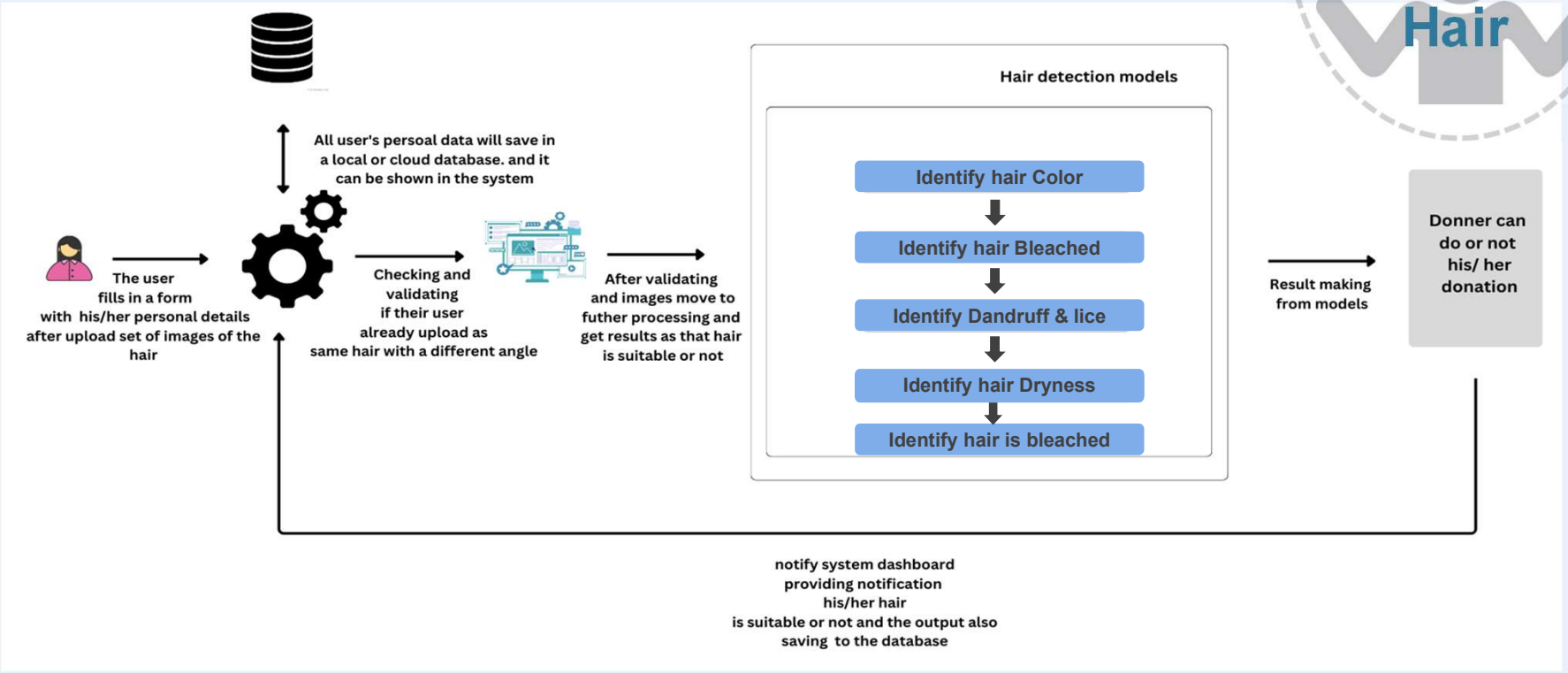
Sub Objectives

01 - Collected data which needs to train the prediction models

02 - Combine all the models to implement the final model

03 - Implement the Web Application with User-friendly Interface.

Component Diagram



Methodology

- Data collection
  - Data Pre Processing and Create Data Set With Data Augmentation
- Used transfer learning based Convolutional Neural Network(CNN) architectures for training models
  - Data augmentation and selected the best technique to retrain models to achieve good fitting
- Tuning hyper-parameters and retrained models to select the best architecture
  - Data Pre Processing and Create Data Set With Data Augmentation
- Data Visualization
  - Host the finalized model in the flask server
- Displayed the results in web application

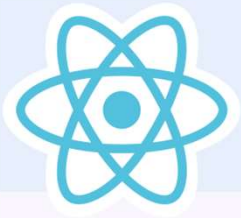
Used Techniques and Technologies

Techniques

- Transfer learning
- Data Augmentation
- Normalization

Technologies

- React
- Python
- TensorFlow
- Anaconda Environment
- Node Server
- Jupyter Notebook
- Google Colab
- Visual Studio Code





Preprocessing for the image data sets

In [4]:
```
import cv2
import imghdr
```

In [6]:
```
image_exts = ['jpeg','jpg', 'bmp', 'png']
```

In [7]:
```
data_dir='C:/Users/Ridma/Downloads/Hairs'
for image_class in os.listdir(data_dir):
    for image in os.listdir(os.path.join(data_dir, image_class)):
        image_path = os.path.join(data_dir, image_class, image)
        try:
            img = cv2.imread(image_path)
            tip = imghdr.what(image_path)
            if tip not in image_exts:
                print('Image not in ext list {}'.format(image_path))
                os.remove(image_path)
        except Exception as e:
            print('Issue with image {}'.format(image_path))
            # os.remove(image_path)
```

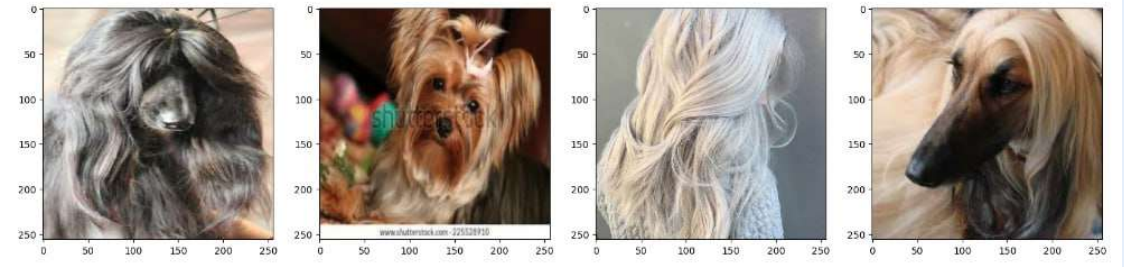
In [ ]:
```
datagen = preprocessing.image.ImageDataGenerator(
    rescale=1./255,
    validation_split=0.2
```

01.Model for the Hair identification

The donor will confirm whether the images uploaded to the system are related to a hair or not. It was found that **MobileNetV2** is the best architecture to implement in image processing.

- To implement this model the data was collected through istockphoto data collection.

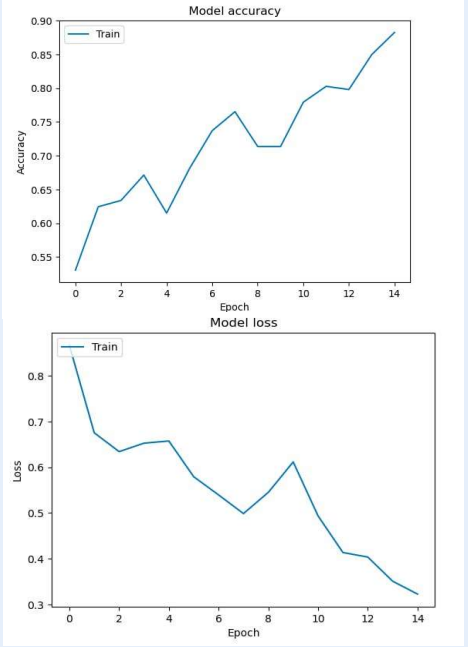
```
for idx, img in enumerate(batch[0][:4]):
    ax[idx].imshow(img.astype(int))
```



Proposed Model

Classes	Train and Validation data
Hair	Train - 460 Validation -110
Others	Train - 440 Validation - 108

Data Collection



Accuracy: 0.9560

Loss: 0.0453

Best Architecture

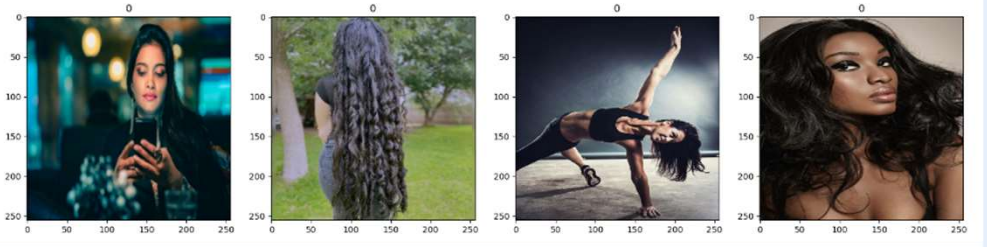
MobileNetV2

02.Model for the Hair Color Detection

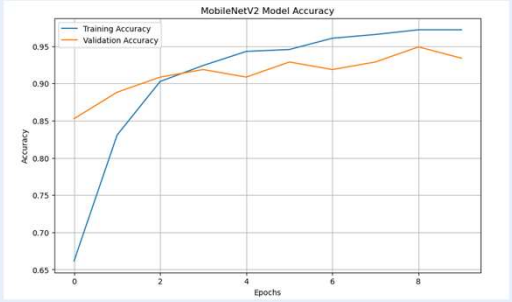
According to the hair donation rules of Apeksha Hospital, currently they only receive black hair. Other colors hair is not accepted. This system uses a model to identify the color of the donor's hair. Only black color hair can be donated here.

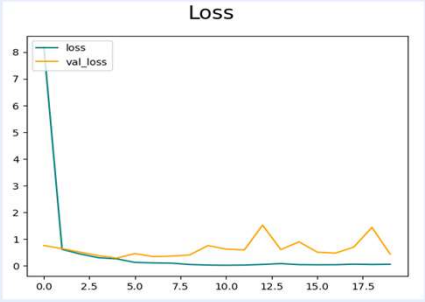
- To implement this model the data was collected through istockphoto data set.

```
In [5]: data = tf.keras.utils.image_dataset_from_directory('C:/Users/Ridma/Downloads/newimages')
        Found 989 files belonging to 2 classes.
In [6]: data_iterator = data.as_numpy_iterator()
In [7]: batch = data_iterator.next()
In [8]: fig, ax = plt.subplots(ncols=4, figsize=(20,20))
        for idx, img in enumerate(batch[0][:4]):
            ax[idx].imshow(img.astype(int))
            ax[idx].title.set_text(batch[1][idx])
```



Proposed Model





Classes	Train and Validation data
Black Hair	Train - 480 Validation -100
Other colours Hair	Train - 450 Validation - 105

Data Collection

Loss: 0.0576

Accuracy: 0.9785

Best Architecture

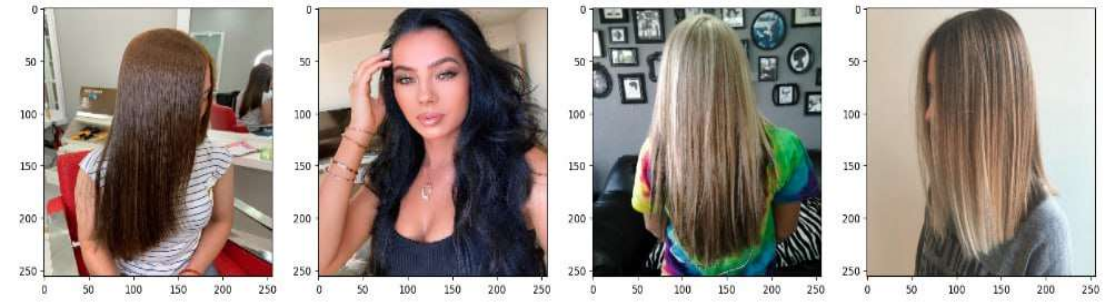
MobileNetV2

03.Model for the hair Bleached Detection

It verifies whether the hair images uploaded by the donor are beached or not.

- To implement this model the data was collected through a Kaggle data set.

```
fig, ax = plt.subplots(ncols=4, figsize=(20, 20))
for idx, img in enumerate(batch[0][:4]):
    ax[idx].imshow(img.astype(int))
```



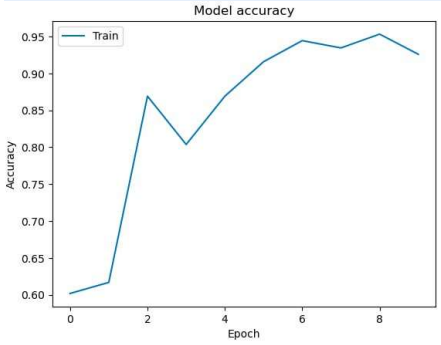
Proposed Model

Classes	Train, Validation and Test data
bleached Hair	Train - 150 Validation - 37
Not bleached Hair	Train - 430 Validation - 76

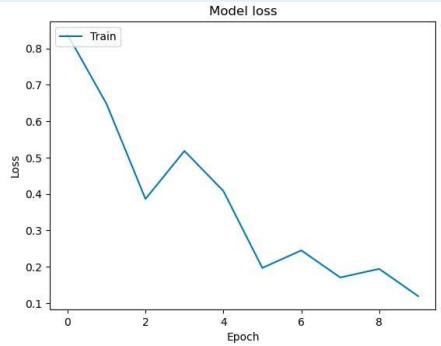
Best Architecture

VGG16

Data Collection



Accuracy: 0.9345



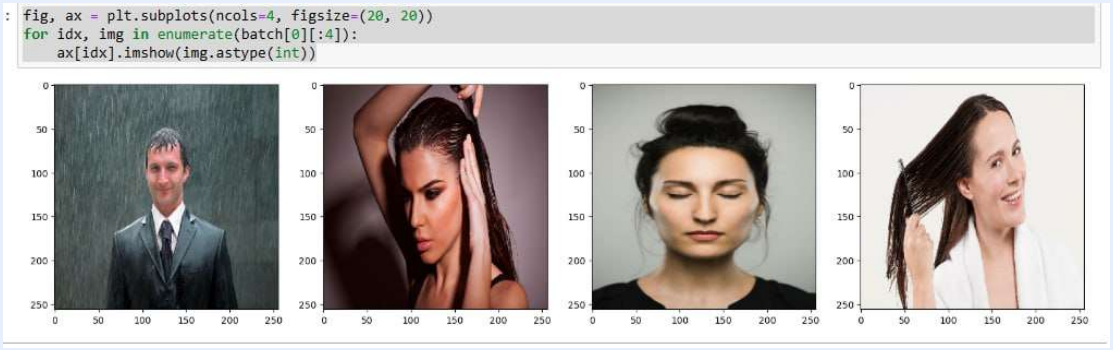
Loss: 0.2021

04.Model for the hair Dryness Detection

The hair images uploaded by the donor will confirm whether the hair is dry or not.

- To implement this model the data was collected through a Kaggle data set.

```
fig, ax = plt.subplots(ncols=4, figsize=(20, 20))
for idx, img in enumerate(batch[0][:4]):
    ax[idx].imshow(img.astype(int))
```



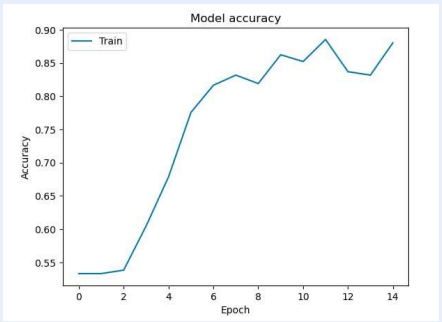
Proposed Model

Classes	Train, Validation and Test data
Dry Hair	Train - 468 Validation - 67
Wet Hair	Train - 130 Validation - 36

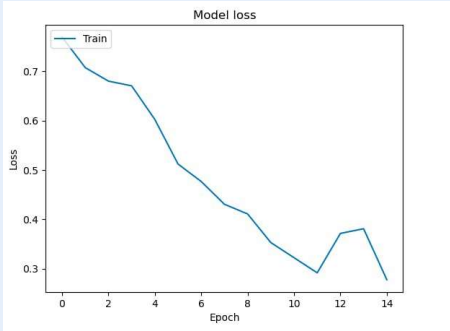
Best Architecture

VGG16

Data Collection



Accuracy: 0.8856



Loss: 0.2002

05.Model for the Dandruff & Lice Detection

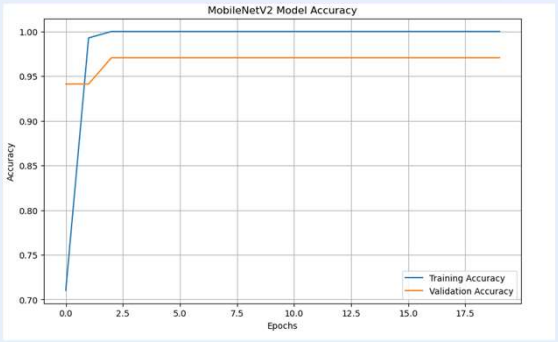
As per hair donation rules at Apeksha Hospital, they require clean, healthy hair free from dandruff and lice infestation. This system checks whether the donor's hair is healthy and free from dandruff and lice.

How We can Detect Hair Quality

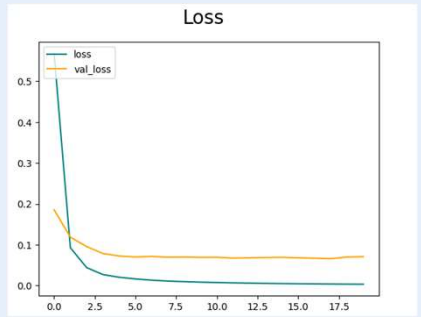
- To implement this model the data was collected through istockphoto data set.

```
In [12]: data_iterator = data.as_numpy_iterator()

In [13]: batch = data_iterator.next()

In [14]: fig, ax = plt.subplots(ncols=4, figsize=(20,20))
         for idx, img in enumerate(batch[0][:4]):
             ax[idx].imshow(img.astype(int))
             ax[idx].title.set_text(batch[1][idx])
```

Proposed Model



Accuracy: 1.0000



Loss:0.0032

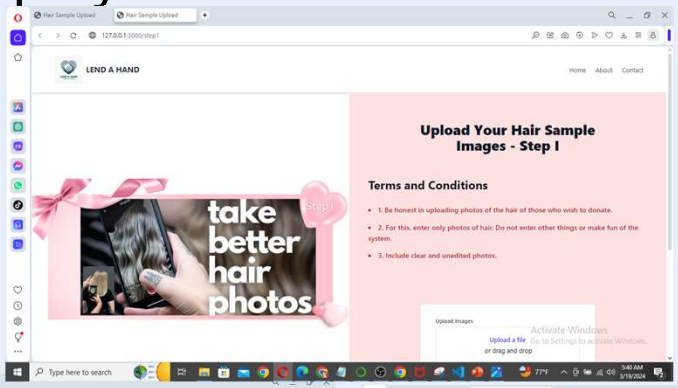
Classes	Train, Validation and Test data
Lice & Dandruff Hair	Train - 160 Validation -48
Clean Hair	Train - 150 Validation - 40

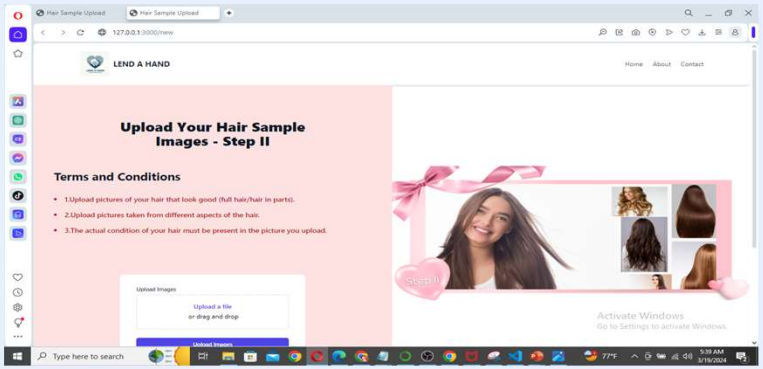
Data Collection

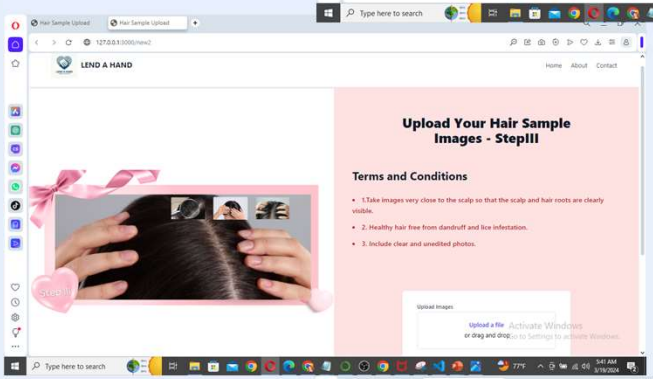
Best Architecture

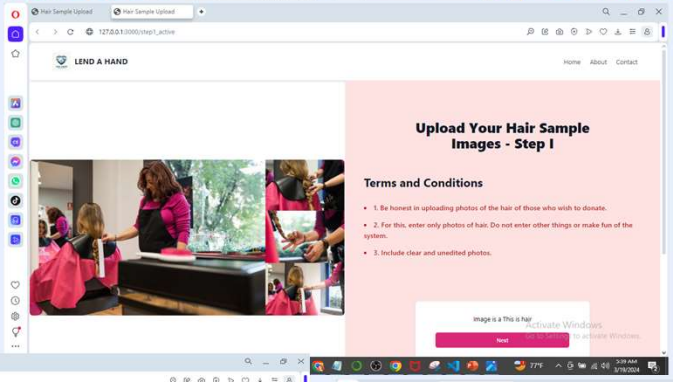
MobileNetV2

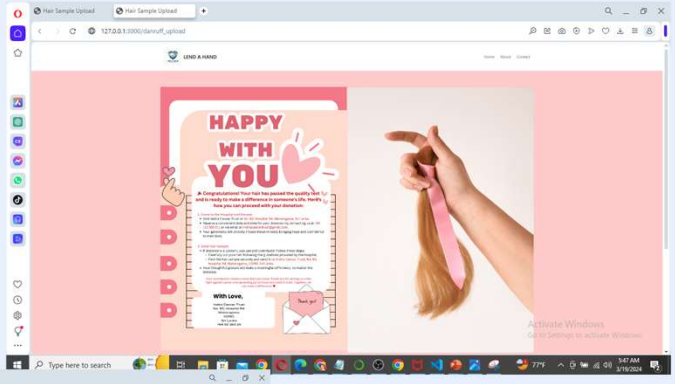
Display the results in Web UI

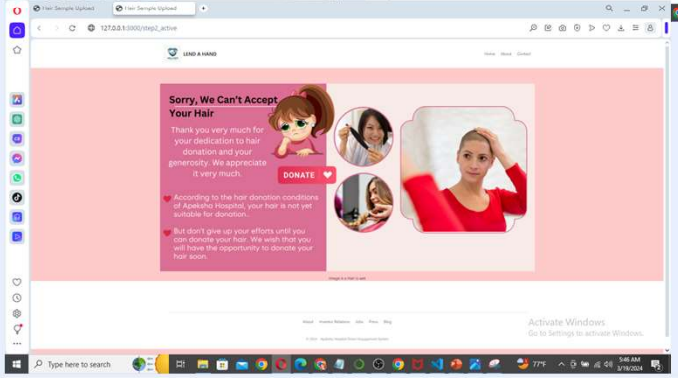


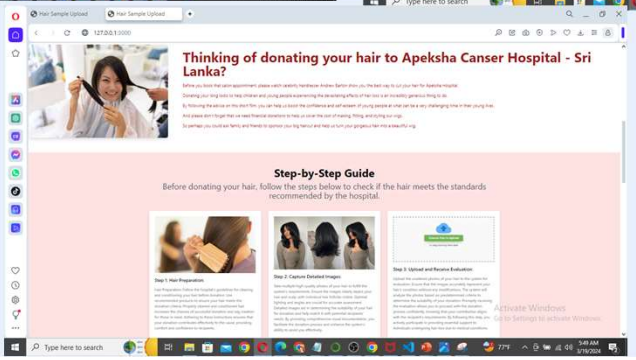




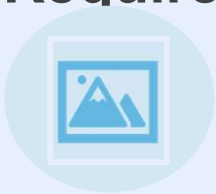








Functional Requirements



•VERIFY THAT THE PHOTOGRAPHS PROVIDED BY THE DONORS ARE OF THE HAIR

•GET EXACT MATCH OF DONOR PROVIDED PHOTOS WITH THE FACTORS CHECKED BY THE SYSTEM.

•ACCORDING TO THE FINAL OUTPUT OF THE SYSTEM, IT MUST BE CORRECTLY IDENTIFIED WHETHER IT IS SUITABLE OR UNSUITABLE HAIR FOR DONATION.

Non-Functional Requirements

- Accuracy
- Usability
- Availability
- User-friendliness
- Efficiency

Completion and Future works

Completion of the components

- Collected data which needs to train the prediction models - Sub Objective 1
- Identification of best architecture for transfer learning
- Model Implemented to Hair identification– Main Objective 1.2
- Model Implemented to identify Color of Hair – Main Objective 1.3
- Model Implemented to identify Dandruff & lice of Hair - Main Objective 1.4
- Model Implemented to identify Dryness of Hair - Main Objective 1.5
- Model Implemented to identify bleached of Hair - Main Objective 1.6
- Combine all the models to implement the final model - Sub Objective 2
- Implement the Web Application with User-friendly Interface - Sub Objective 3

Future Implementations

- System Integration with other functions
- Host the finalized models