2023-24-100

Apeksha Hospital Donor Engagement system

SLIIT
FACULTY OF COMPUTING

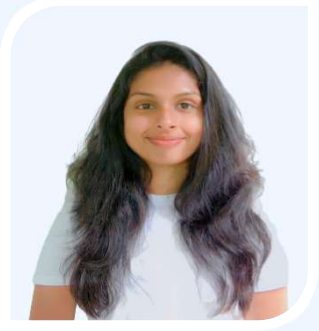
20/05/2024

Presented by

Punchihewa S.N
IT20665166

Prabodha K.W.D.S
IT20665098

Bandara H.R.H.S
IT20665166



Wijesooriya P.L.P.G
IT20660352

Ms. Lokesha Weerasinghe
Supervisor

Ms. Chamali Pabasara
Co-Supervisor

SLIIT
FACULTY OF COMPUTING

20/05/2024

Introduction

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

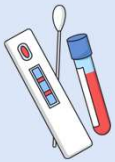
Research Problem

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.

Components

01 Intelligent Donor-Driven Inventory System for Essential Items

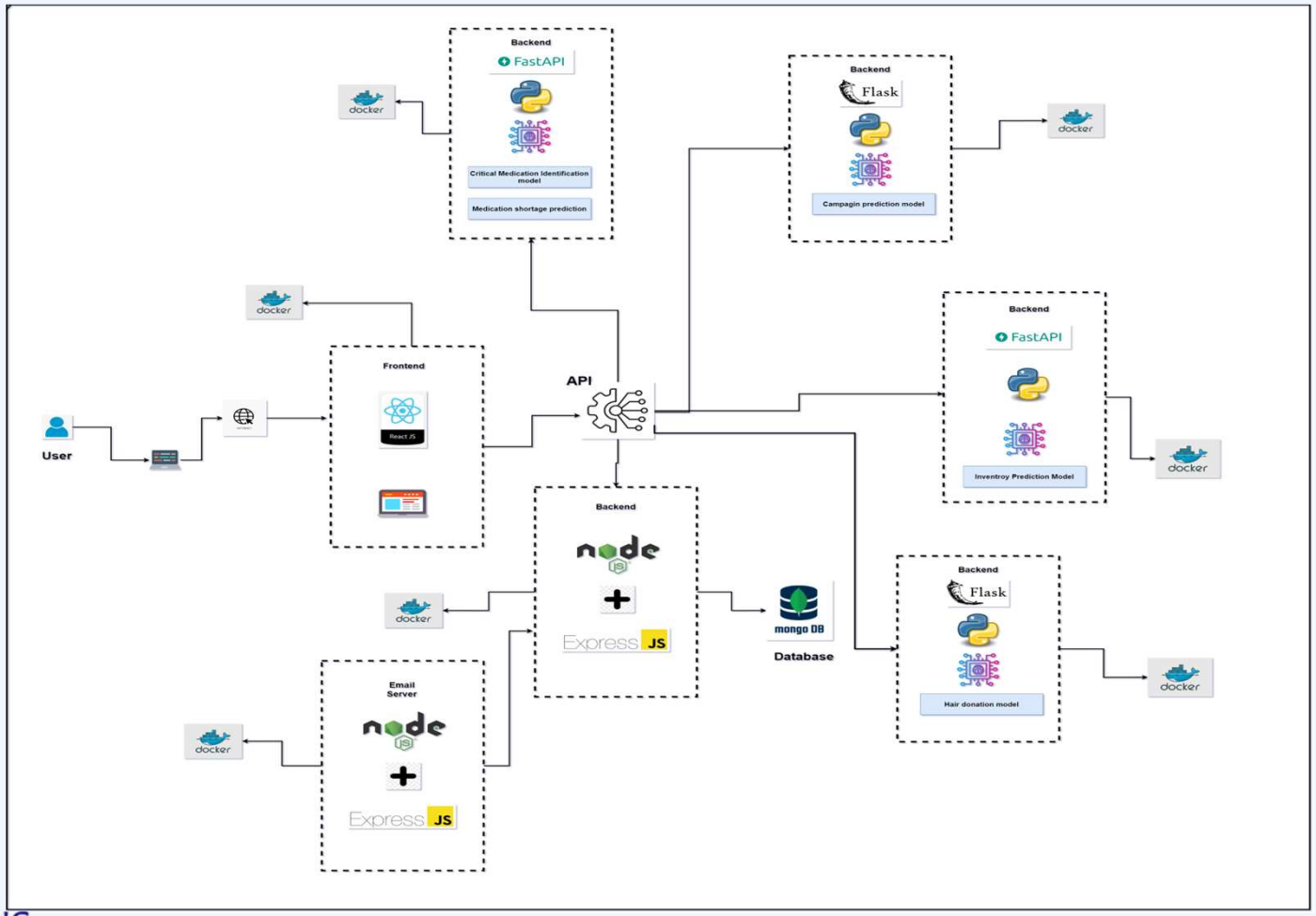
02 Critical Medication Priority Recommender System



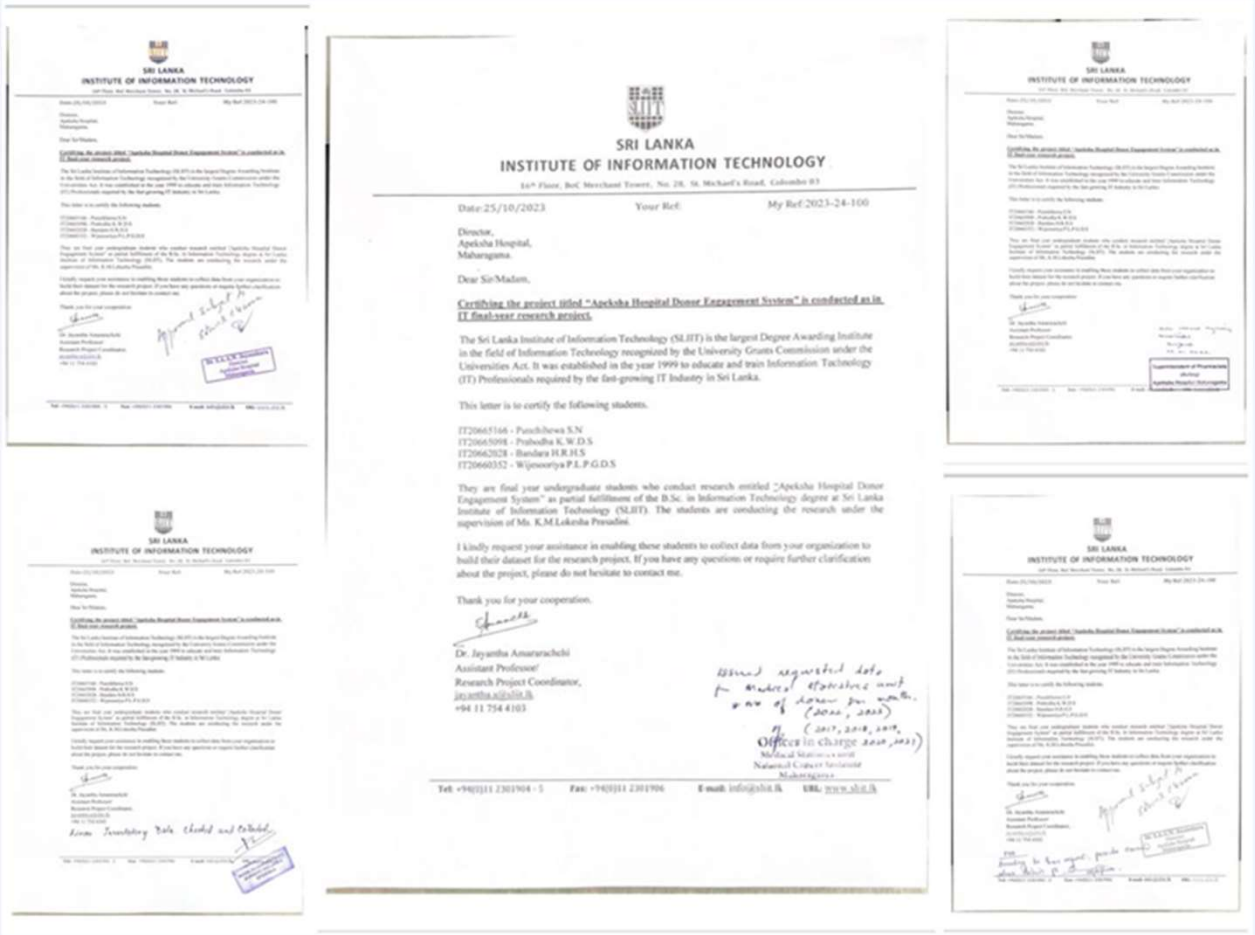
03 Predictive Analytics for Donation Campaign Success

04 Promoting quality hair donation for cancer patients

System Overview Diagram



Data Collection Evidence



SRI LANKA
INSTITUTE OF INFORMATION TECHNOLOGY
16th Floor, Bod 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 - Panchibewa S.N
IT20665098 - Prabodha K.W.D.S
IT20662928 - 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.

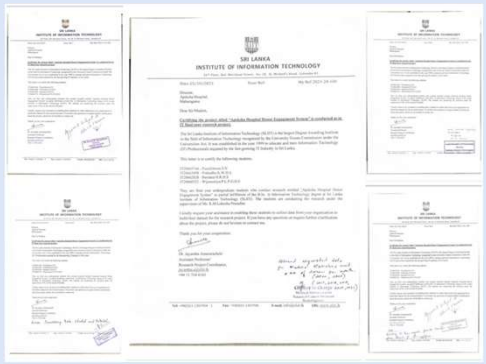
I kindly request your assistance in enabling these students to collect data from your organization to build their dataset for the research project. If you have any questions or require further clarification about the project, please do not hesitate to contact me.

Thank you for your cooperation.

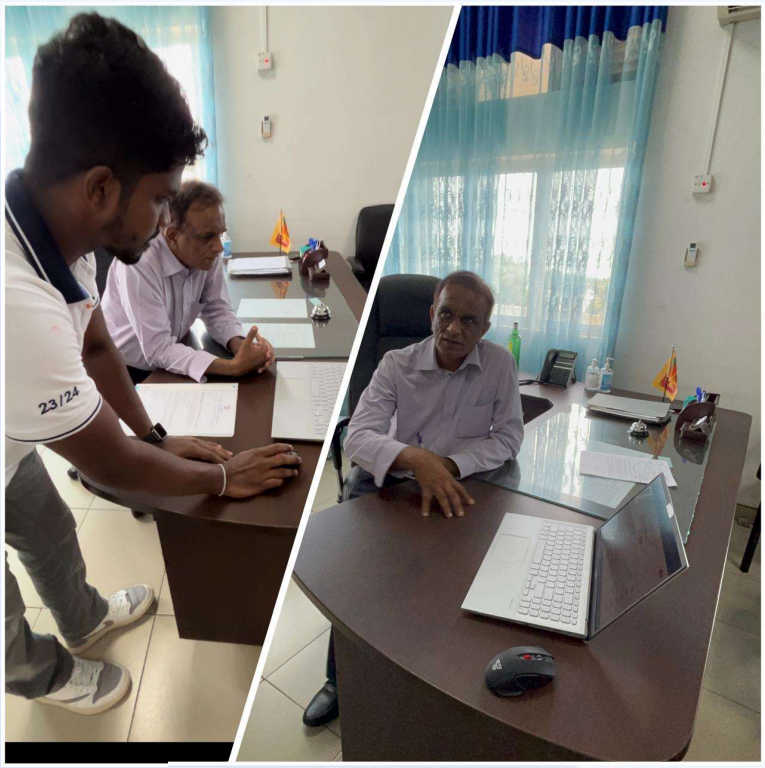
Dr. Jayantha Amararachchi
Assistant Professor/
Research Project Coordinator,
jayantha.a@sliit.lk
+94 11 754 4103

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

Stakeholder feedback



Data Collection Evidance



Getting feedbacks from
Dr. Roshan Amaratunga

(Acting Director, Apeksha Hospital)

Commercialization

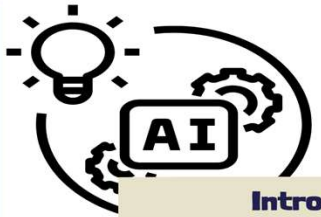
Enhanced resource management leads to better patient outcomes, making the system attractive to healthcare providers focused on quality care.

- Scalability to other hospitals
- Customization for different healthcare settings

By optimizing resource allocation and reducing wastage, the system can lower operational costs for hospitals

- Partnership and collaboration opportunities
- Training and support services

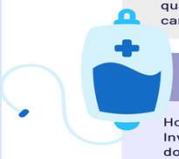
Poster



DONOR MANAGEMENT AND SCARCE MEDICAL RESOURCE ALLOCATION IN A CANCER HOSPITAL IN SRI LANKA

Introduction

Apeksha Hospital's Intelligent Donor-Driven Inventory System uses machine learning to match donations with needs, including critical medication prioritization, blood donation campaigns, and hair quality assessment for donations, enhancing patient care.

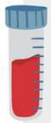


Research Question

How can Apeksha Hospital's Intelligent Donor-Driven Inventory System, using machine learning, optimize donor contributions to meet real-time needs, enhance critical medication management, and improve blood and hair donation processes?

Methodology

This study uses machine learning to predict medical resource needs at Apeksha Hospital. Data collection, preprocessing, and feature extraction are conducted on hospital records. Decision Tree Regression is used for inventory prediction, Random Forest for medication priority, and predicting donation campaign success. Image processing techniques are employed to assess hair quality for donations. Evaluation metrics include accuracy, precision, and recall, ensuring efficient resource allocation for inventory, medication, blood, and hair donations.



Findings

The implementation of machine learning algorithms in predicting medical resource needs yielded promising results. Decision Tree Regression accurately forecasted inventory requirements, optimizing resource utilization. Random Forest models effectively identified medication shortages and prioritized critical medications with high accuracy, enhancing patient care quality. Predictive analytics successfully forecasted blood donation campaign success, aiding in strategic planning for donation initiatives. Additionally, image processing techniques facilitated the assessment of hair quality for donations, streamlining the hair donation process. Overall, these findings demonstrate the efficacy of data-driven approaches in healthcare management, ensuring efficient resource allocation and improving patient outcomes.

Conclusion

Machine learning integration at Apeksha Hospital improved resource management and patient care. Accurate inventory forecasting and medication prioritization optimized allocation. Predictive analytics boosted blood donation campaigns, ensuring continuous supply. Image processing streamlined hair donation, enhancing donation management. These findings highlight data-driven healthcare's pivotal role in efficient resource utilization and improved patient outcomes.

Results

The implementation of machine learning algorithms yielded promising outcomes. Decision Tree Regression accurately forecasted inventory requirements, optimizing resource allocation. Random Forest models effectively identified medication shortages and prioritized critical medications, enhancing patient care. Predictive analytics successfully forecasted blood donation campaign success, aiding strategic planning. Image processing techniques streamlined hair donation assessments. These results underscore the efficacy of data-driven approaches in healthcare management, improving resource allocation and patient outcomes.

IT20665166
Punchihewa S.N

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

SLIIT
FACULTY OF COMPUTING

20/05/2024

Introduction

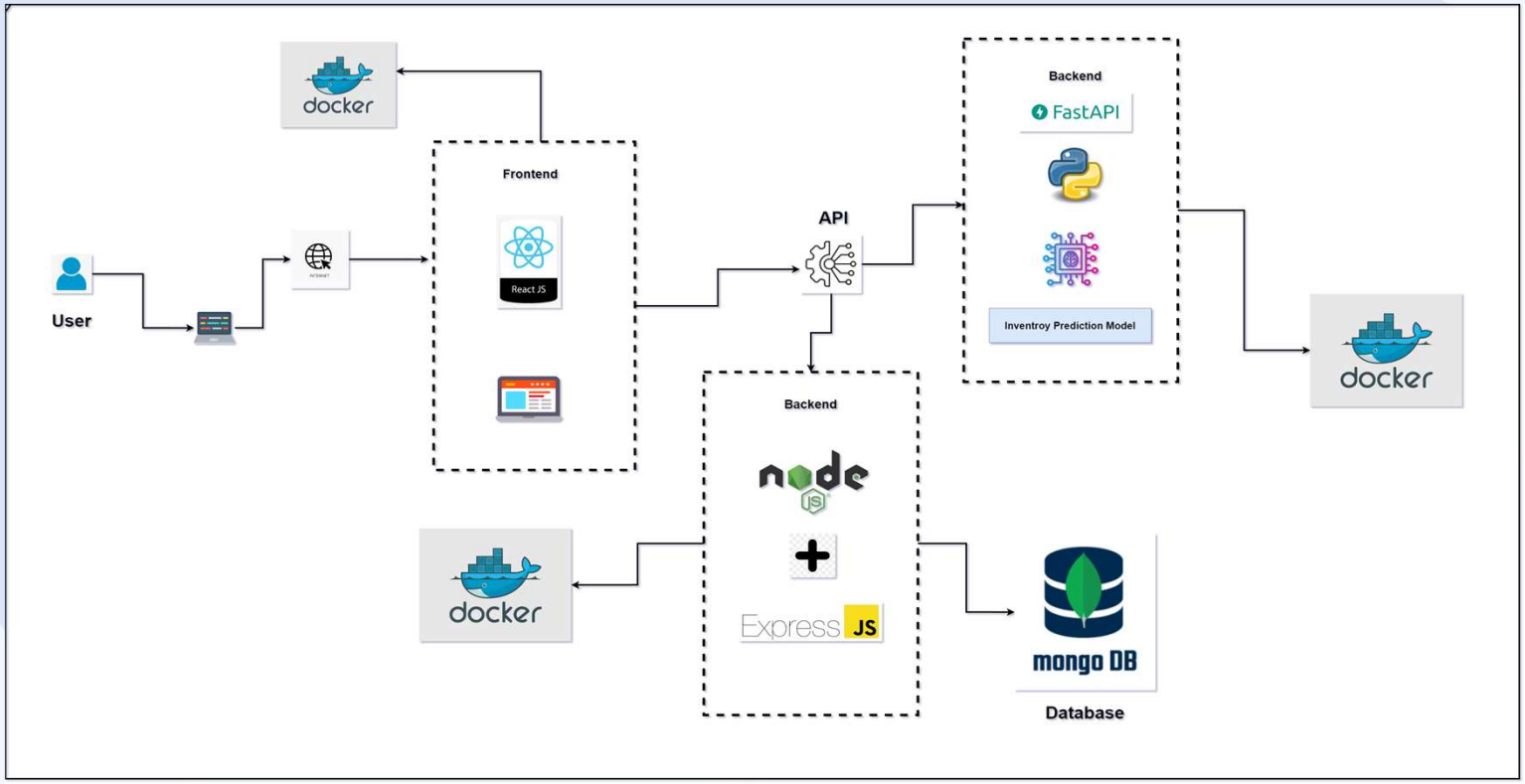
Apeksha Hospital faces donor engagement and inventory challenges. We propose a machine learning system to recommend needed items by analyzing donor behavior. This platform aligns donor preferences with hospital needs, enhancing satisfaction and efficiency.

Research 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.

Methodology

Intelligent Donor-Driven Inventory System for Essential Items



Technologies and Techniques

Programming Language

- Python
- JavaScript

Tools

- Jupyter Notebook
- Scikit Learn
- Anaconda Navigator
- FastAPI
- Render

Algorithm

- Decision Tree Regression

Version Controlling

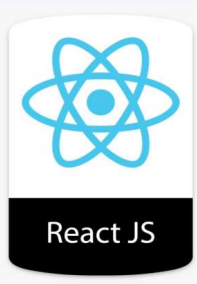
- GitHub

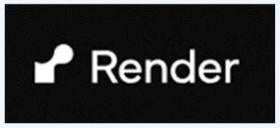
Containerization

- Docker

Framework

- NodeJS
- ReactJS





Objectives

Intelligent Donor-Driven Inventory System for Essential Items

Data Collection

- ❖ **Collected data from Apeksha Hospital**

Data Pre-Processing

- ❖ **Conversion of Categorical Variables to Numerical format**

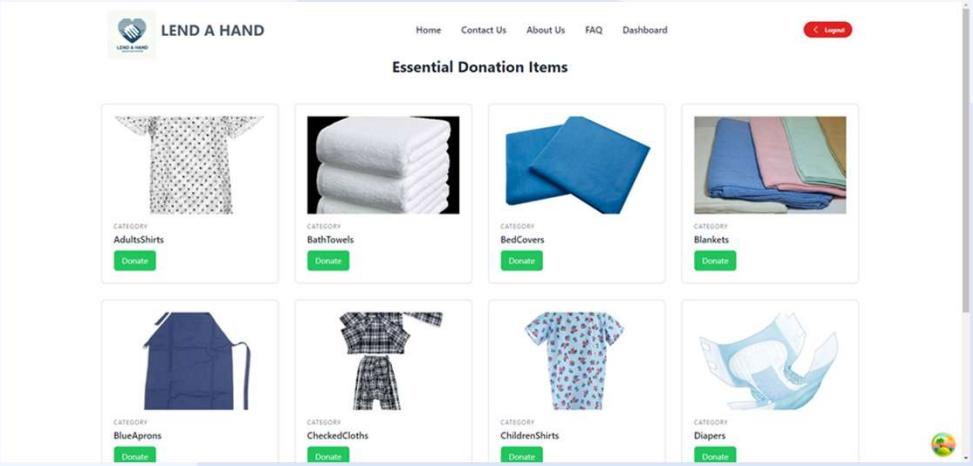
Implement Prediction Model

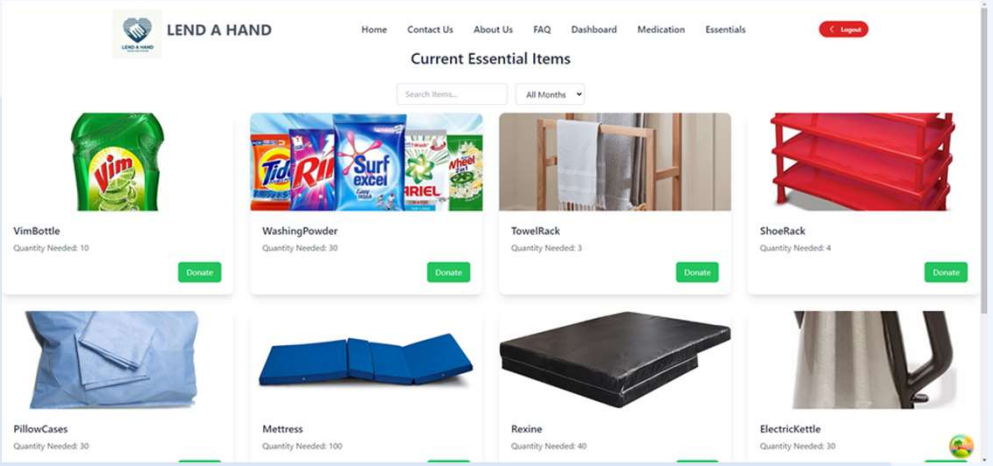
- ❖ **Implement Machine Learning Model Using "Decision Tree Regression "**

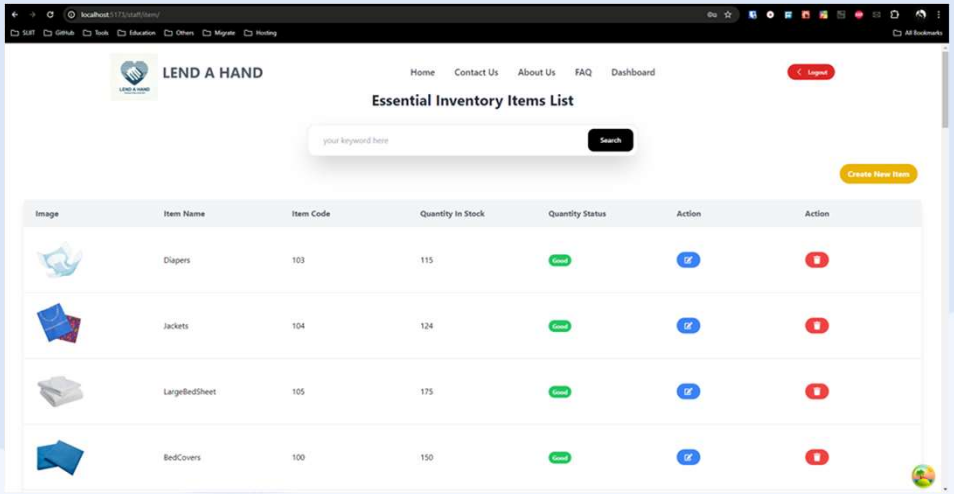
```
[2]: import pandas as pd
     from sklearn.model_selection import train_test_split
     from sklearn.tree import DecisionTreeRegressor
     from sklearn.preprocessing import LabelEncoder

[3]: # Load the dataset
     df = pd.read_csv('newdataset.csv')

[4]: # Preprocess the Data
     le_ItemName = LabelEncoder()
     le_Month = LabelEncoder()
     df["ItemName"] = le_ItemName.fit_transform(df["ItemName"])
     df["Month"] = le_Month.fit_transform(df["Month"])

[5]: # Define a dictionary to map encoded values to original item names
     item_name_mapping = dict(zip(le_ItemName.transform(le_ItemName.classes_), le_ItemName.classes_))
     month_name_mapping = dict(zip(le_Month.transform(le_Month.classes_), le_Month.classes_))

[6]: # Define independent variables (features) and the target variable
     X = df[["ItemName", "ItemID", "ItemCategory", "UsageHistory", "Month"]]
     y = df["RequestedQuantity"]

[7]: # Split the data into training and testing sets
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

[8]: # Initialize Decision Tree Regression model
     model = DecisionTreeRegressor(random_state=42)

[9]: # Train the Decision Tree model
     model.fit(X_train, y_train)

[9]: DecisionTreeRegressor
     DecisionTreeRegressor(random_state=42)

[10]: # Make predictions for all items in the dataset
      predictions = model.predict(X)

      # Print the predictions for all items
      for i in range(len(df)):
          item_name_original = item_name_mapping[df.iloc[i]["ItemName"]]  # Map encoded value to original item name
```

```
Month: February
[11]: model.score(X_train, y_train)
[11]: 0.9886072809011376
```

Developed Solution



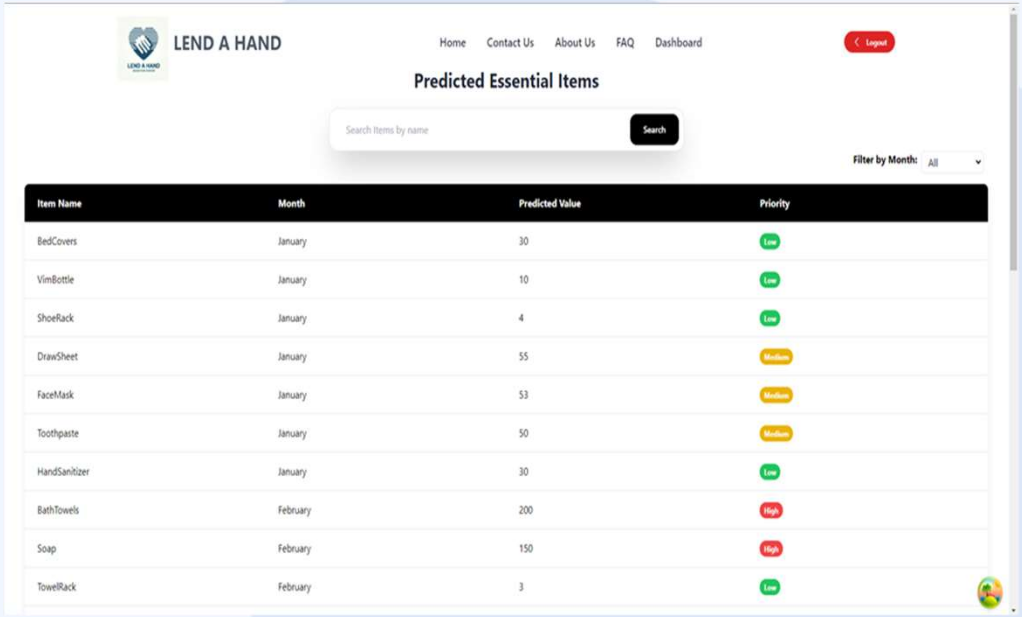


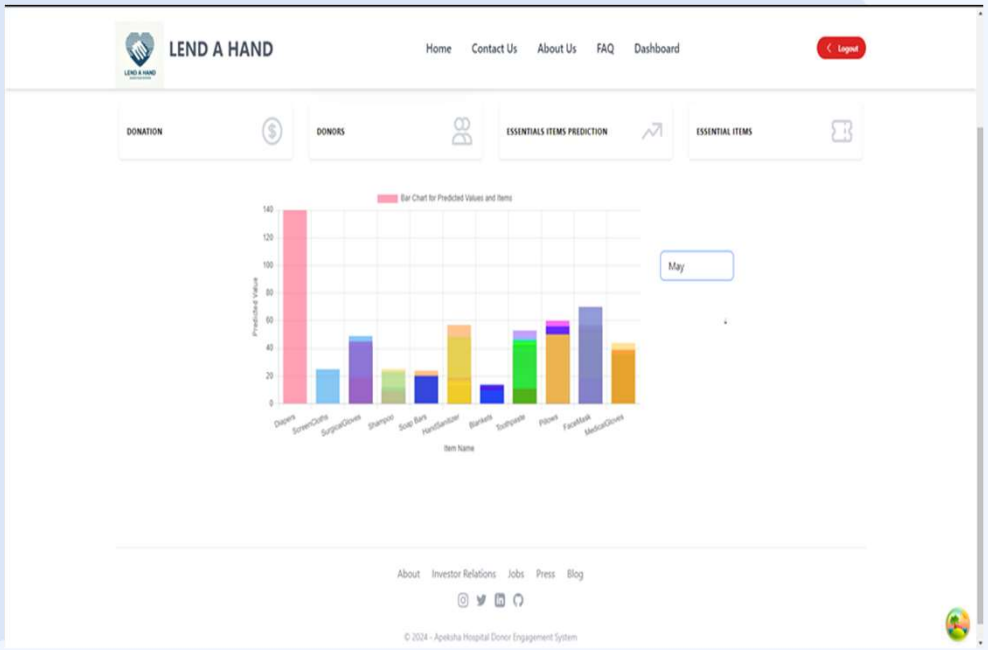


SLIIT FACULTY OF COMPUTING

20/05/2024

Results and Discussion





SLIIT
FACULTY OF COMPUTING

Completeness of the Work

- Identify suitable Machine Learning Algorithm
- Train Machine Learning Model Using Decision Tree Regression
- Predict Inventory Requirements

- Implement Machine Learning Backend Using FastAPI
- Deploy Machine Learning Model using Render Server
- Implement Web Application using MERN Stack Technology
- Component Integration 75% Completed

- Enhance User Interface
- Complete component Integration

IT20665098
Prabodha K.W.D.S

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

SLIIT
FACULTY OF COMPUTING

20/05/2024

Introduction

Critical Medication Priority recommender System

- **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.

Introduction

Critical Medication Priority recommender System

- **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."

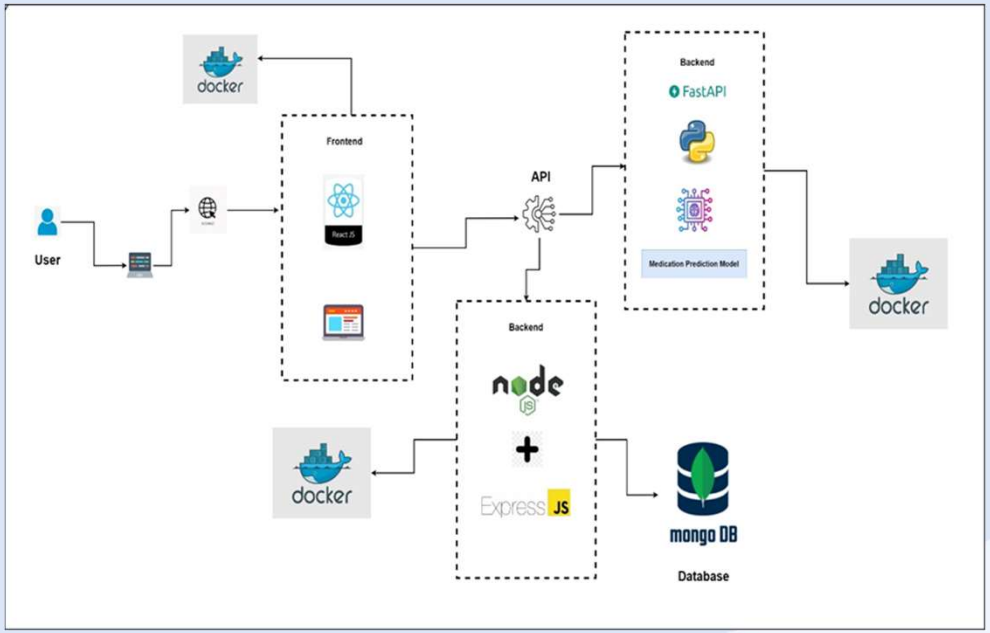
- **Specific Objectives**

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

Methodology

Critical Medication Priority recommender System

- Component Diagram



Methodology

Critical Medication Priority recommender System

- Tools And Technologies

Programming Language

- Python

Tools

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

Version Controlling

- GitHub

Algorithms

- Random Forest Classification



Methodology

Critical Medication Priority recommender System

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

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)

```

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

Methodology

Critical Medication Priority recommender System

Random Forest Classifier **Accuracy- 91%**

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

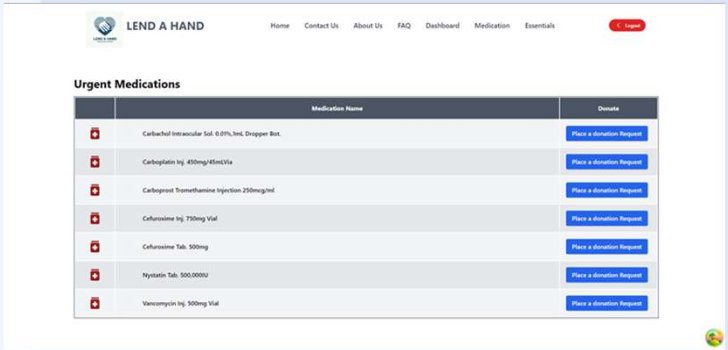
- Implement Machine Learning Model using "**Random Forest**" Algorithm

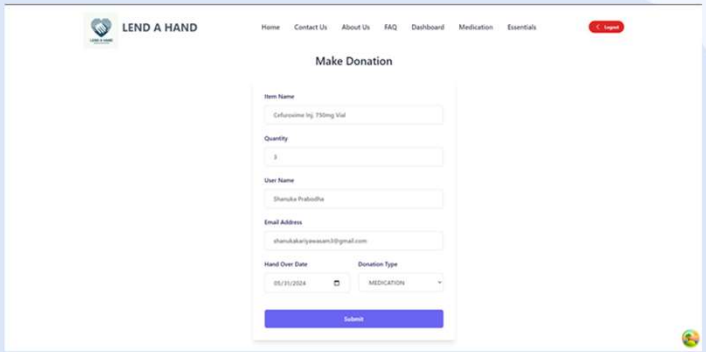
```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestClassifier  # Change
from sklearn.impute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

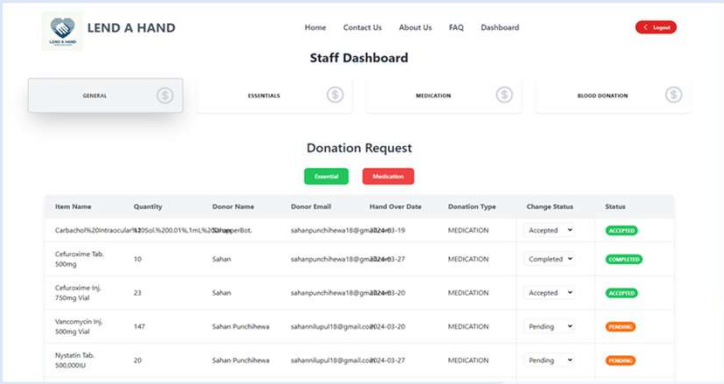
```
      Predicted_Shortage_RF
0                         1
1                         1
2                         1
3                         1
4                         1
...                     ...
1207                      0
1208                      0
1209                      1
1210            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

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

Developed Solution

Critical Medication Priority recommender System

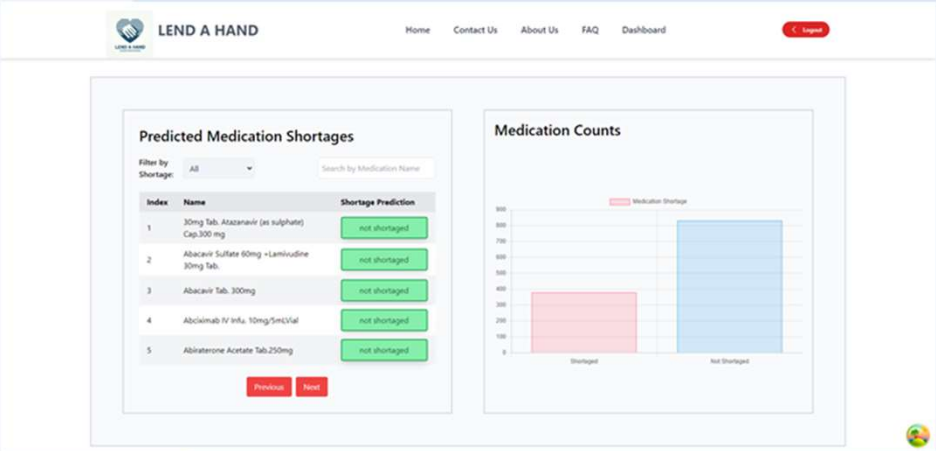


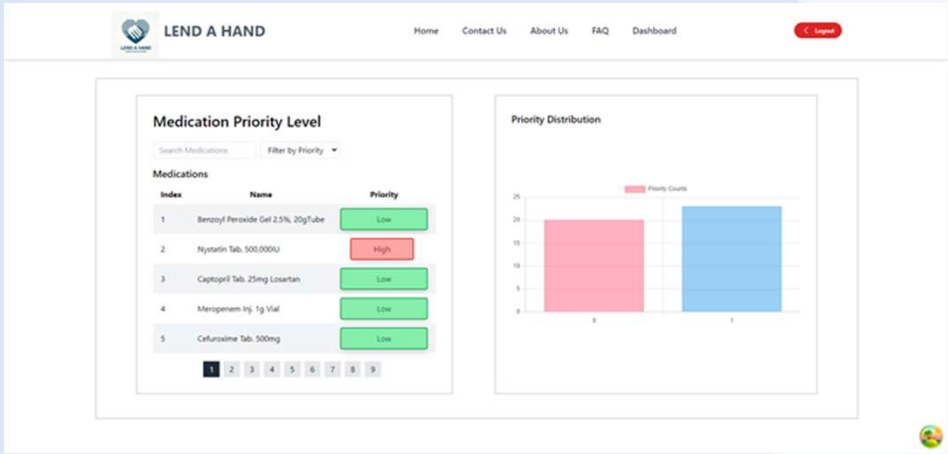




Results and Discussion

Critical Medication Priority recommender System

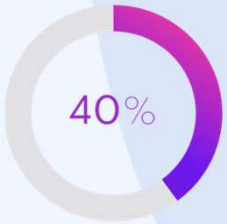




Completeness of the Work

Critical Medication Priority recommender System

- Identify suitable Machine Learning Algorithm
- Train Machine Learning Model Using Random Forest Classifier
- Predict Critical Medications and Medication Shortages
- Implement Machine Learning Backend Using FastAPI
- Deploy Machine Learning Model using Render Server
- Implement Web Application using MERN Stack Technology
- Component Integration 75% Completed



- Enhance User Interface
- Complete component Integration

IT20662028
Bandara H.R.H.S

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

SLIIT
FACULTY OF COMPUTING

20/05/2024

Introduction

Predictive Analytics for Donation Campaign Success

Apeksha Hospital needs a steady blood supply, but our donation campaigns face challenges like unpredictable donor turnout, food wastage, and high marketing costs.

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**

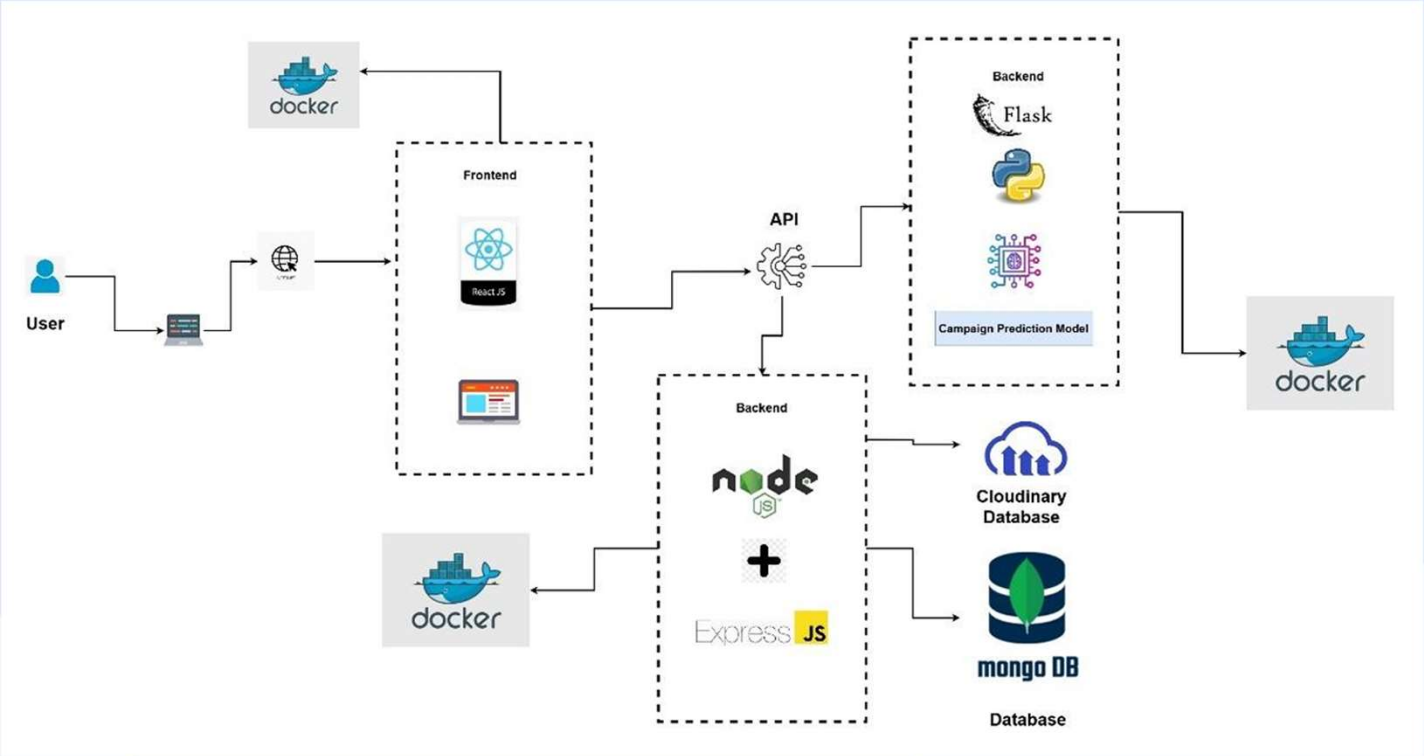
Implement machine learning to enhance the efficiency of blood donation campaigns.

- **Specific Objectives**

1. Optimization Algorithms Integration
2. Predict People amount
3. User-Friendly Interface
4. Manage the limited resources

Methodology

Component Diagram



Technologies and Techniques

Programming Language

- Python
- JavaScript

Tools

- Jupyter Notebook
- Scikit Learn
- Anaconda Navigator
- FlaskAPI
- Render

Algorithm

- Random forest Regression

Version Controlling

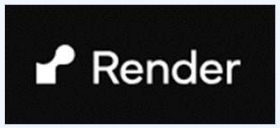
- GitHub

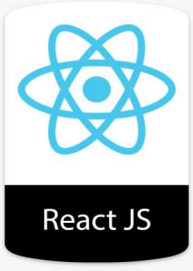
Containerization

- Docker

Framework

- NodeJS
- ReactJS

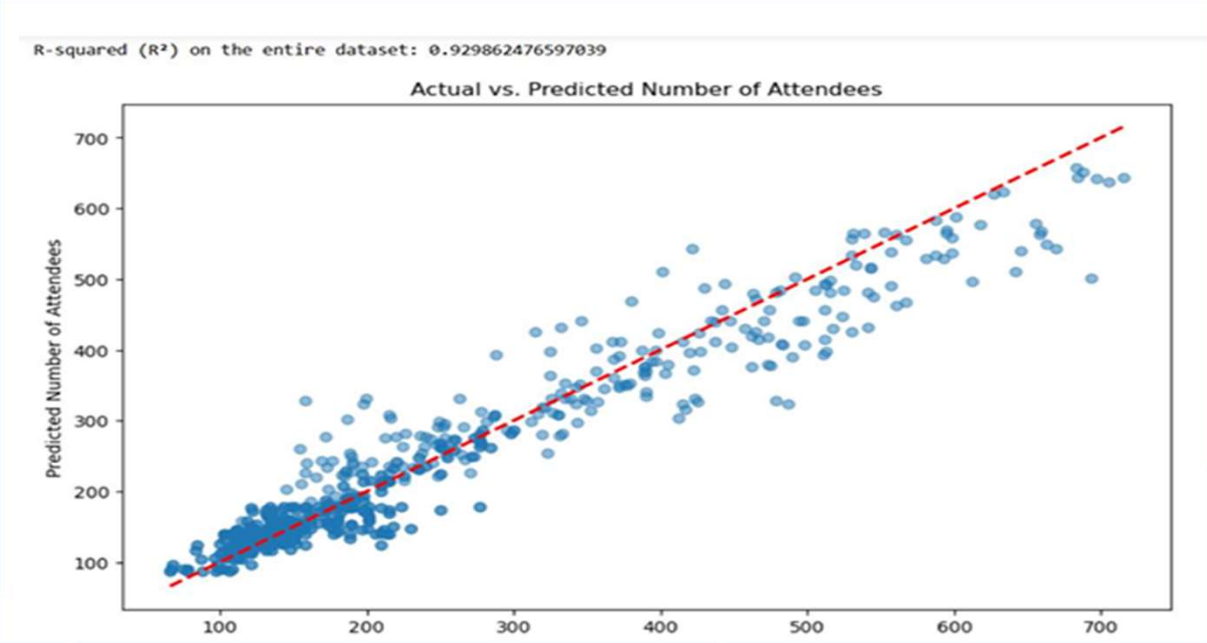




Developed Solution

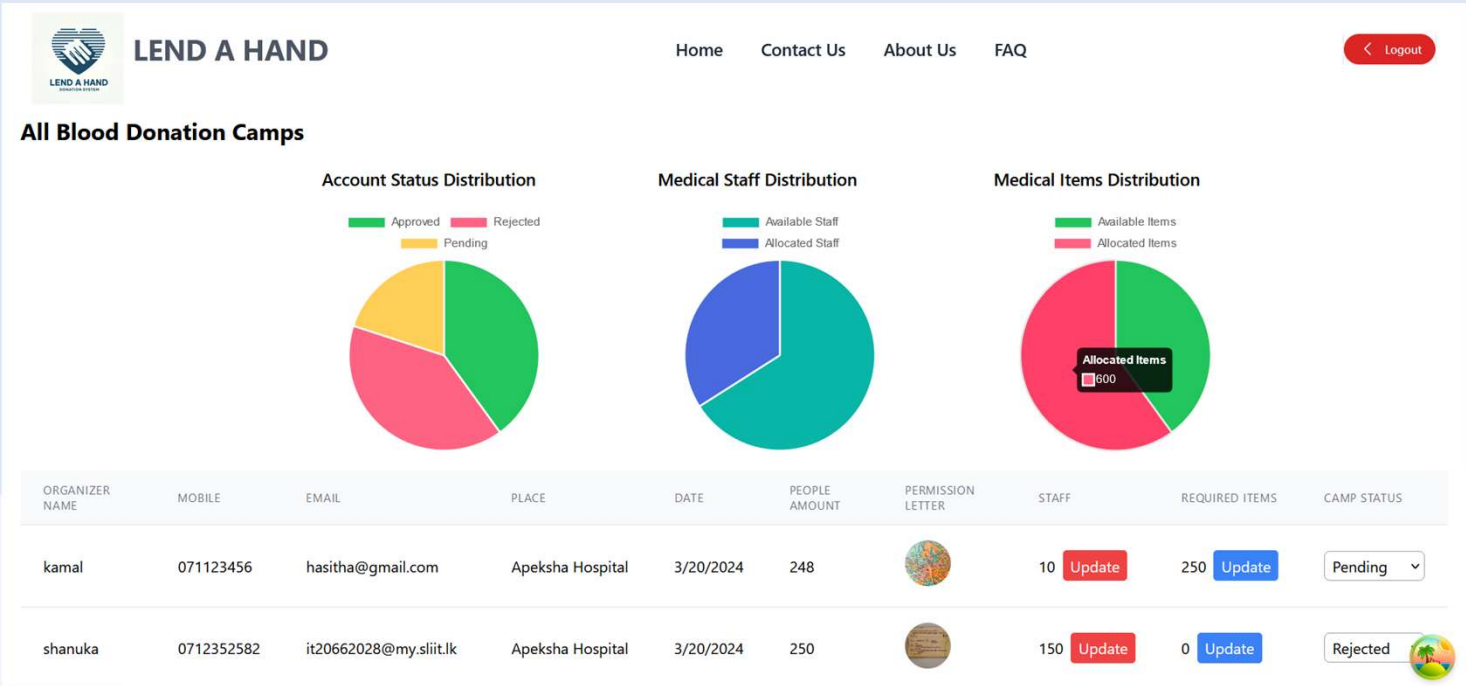
Successfully trained an accurate **machine learning model**

for Predict next campaign people amount - 92% accuracy

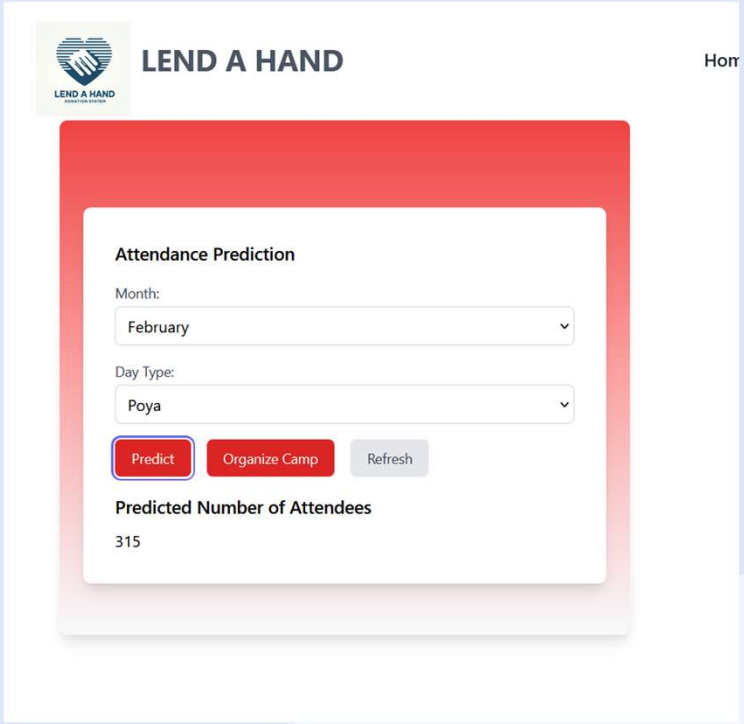


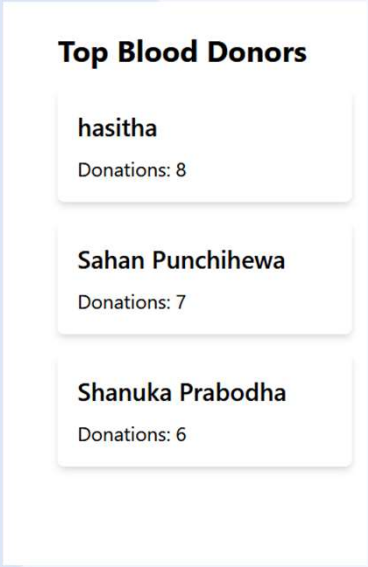
Developed Solution

Manage each donation campaign



Results and Discussion





Completeness of the Work

- Identify suitable Machine Learning Algorithm
- Train Machine Learning Model Using Random forest Regression
- Predict the next campaign people amount

- Implement Machine Learning Backend Using FlaskAPI
- Deploy Machine Learning Model using Render Server
- Implement Web Application using MERN Stack Technology
- Component Integration 75% Completed

- Enhance User Interface
- Complete component Integration

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

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

SLIIT
FACULTY OF COMPUTING

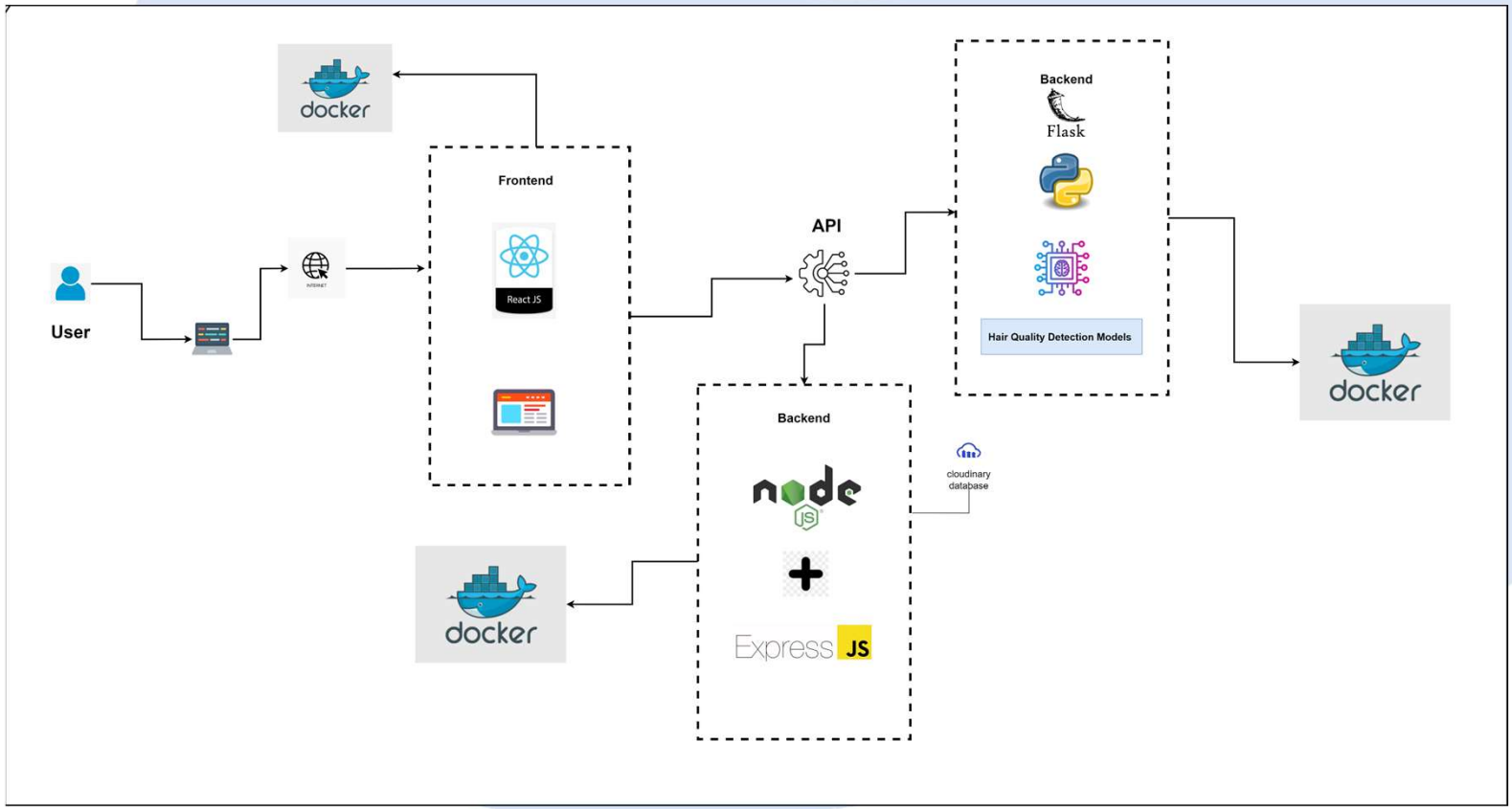
Introduction

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?

Methodology



Technologies and Techniques

Programming Language

- Python
- JavaScript

Tools

- Jupyter Notebook
- Anaconda Navigator
- FlaskAPI
- Render

Architecture

- Convolutional Neural Networks (CNN)

Version Controlling

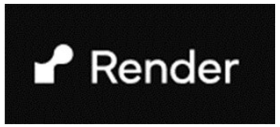
- GitHub

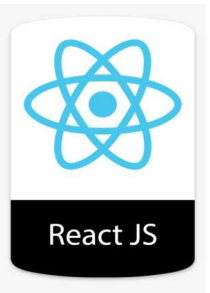
Containerization

- Docker

Framework

- NodeJS
- ReactJS



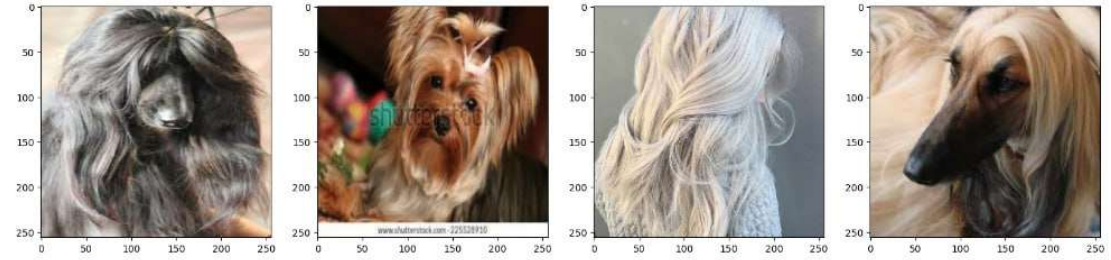


01.Model for the Hair identification

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.

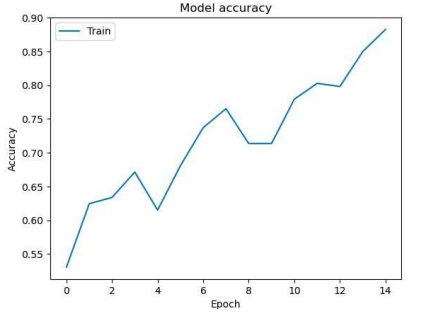
```
for idx, img in enumerate(batch[0][:4]):
    ax[idx].imshow(img.astype(int))
```

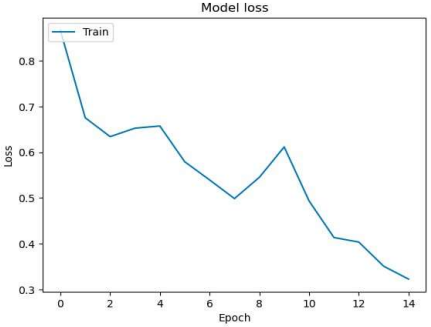


Proposed Model

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

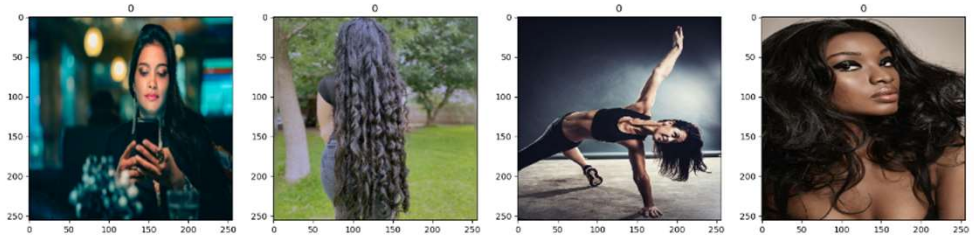
MobileNetV2

02.Model for the Hair Color Detection

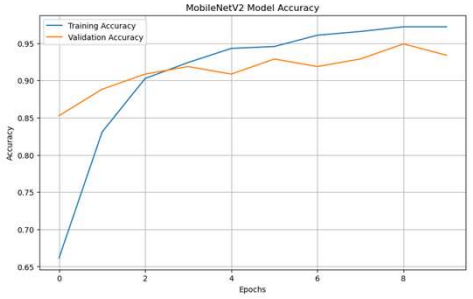
According to the hair donation rules of Apeksha Hospital, currently they only receive black hair. Other colors hair is not accepted. This system uses a model to identify the color of the donor's hair. Only black color hair can be donated here.

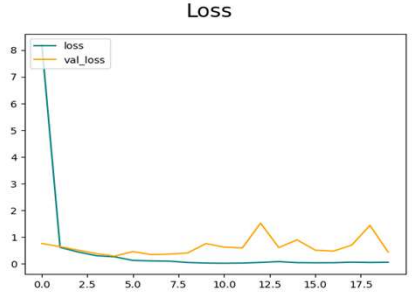
- To implement this model the data was collected through istockphoto data set.

```
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

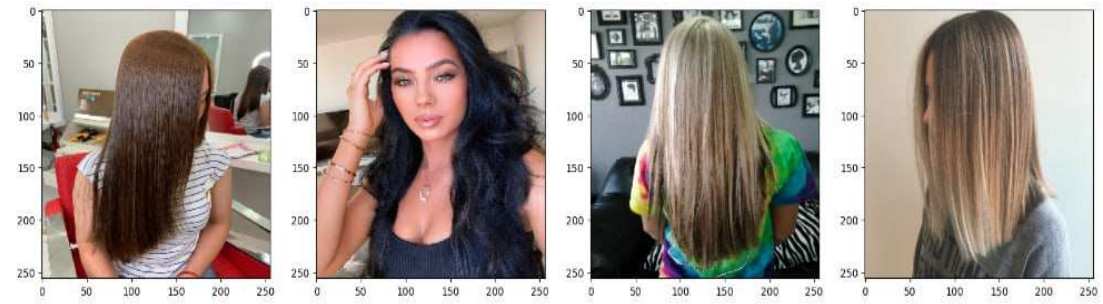
MobileNetV2

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))
```



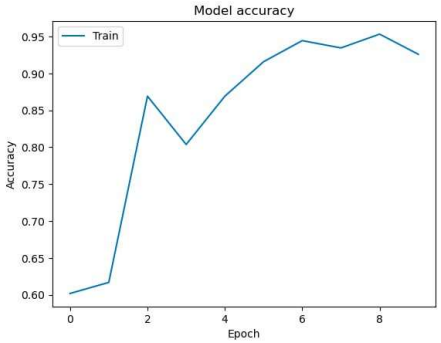
Proposed Model

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

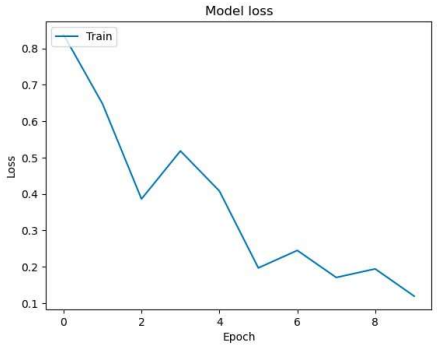
Data Collection

Best Architecture

VGG16



Accuracy: 0.9345



Loss: 0.2021

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

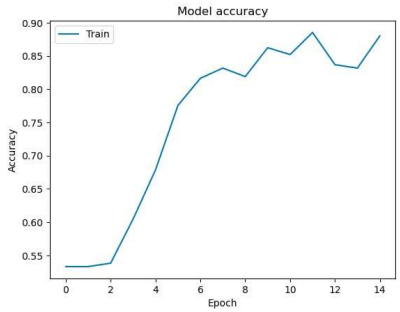
Best Architecture

VGG16

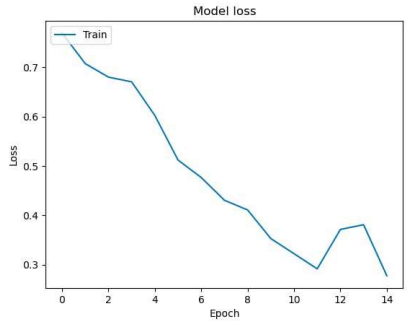
Data Collection

Accuracy: 0.8856

Loss: 0.2002



Accuracy: 0.8856



Loss: 0.2002

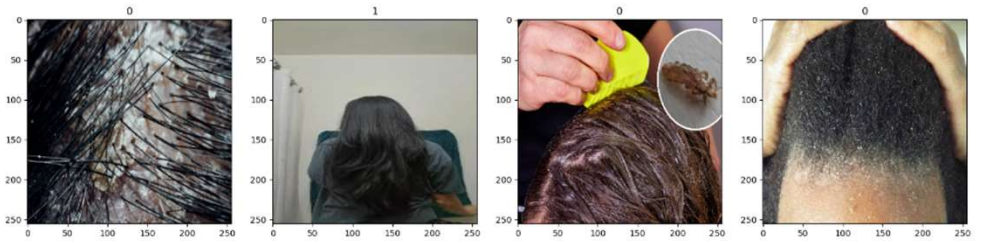
05.Model for the Dandruff & Lice Detection

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.

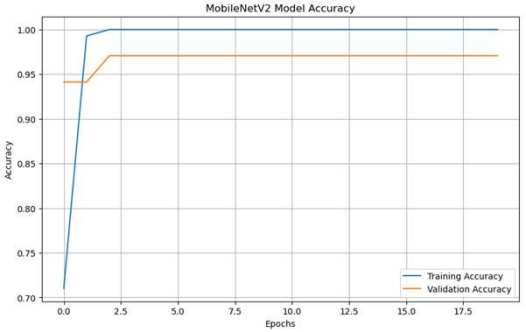
How We can Detect Hair Quality

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

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

[13]: batch = data_iterator.next()

[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])
```

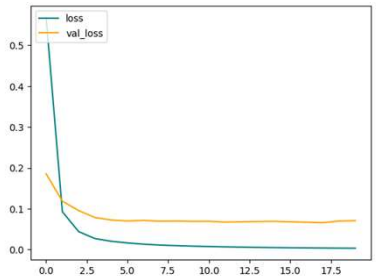


Proposed Model



Accuracy: 9.993

Loss



Loss:0.0032

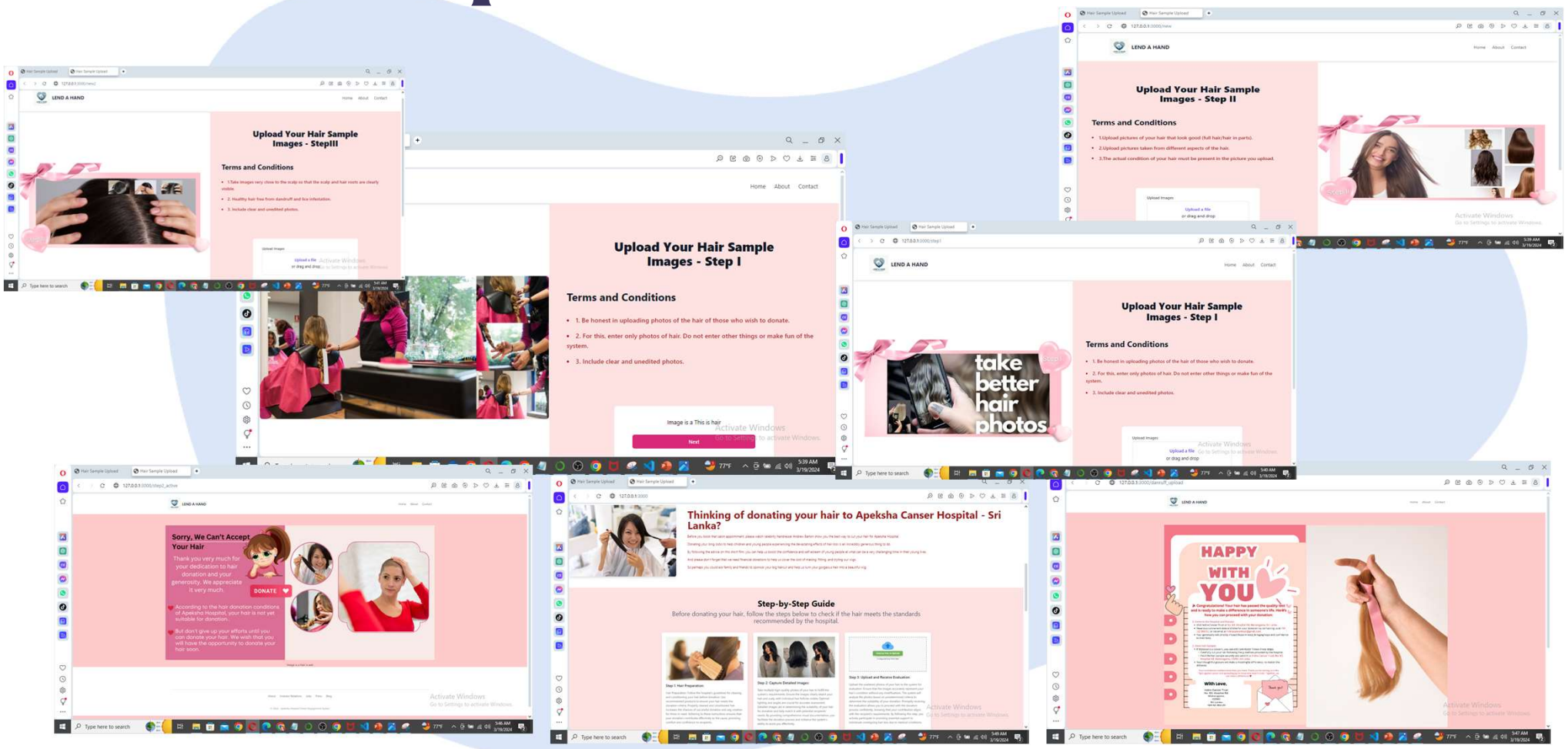
Classes	Train, Validation and Test data
Lice & Dandruff Hair	Train - 160 Validation -48
Clean Hair	Train - 150 Validation - 40

Data Collection

Best Architecture

MobileNetV2

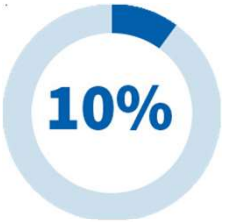
Completeness of the Work



Completeness of the Work

- Data Collection
- Identify suitable Deep Learning Algorithm-CNN
- Train Deep learning Models Using CNN Architectures

- Implement Deep Learning Backend Using FlaskAPI
- Combine All models to implement the final model
- Implement Web Application using MERN Stack Technology
- Component Integration 75% Completed



- Enhance User Interface
- Deploy Deep Learning Models using Render Server
- Complete component Integration

2023-24-100

Thank you for your attention

SLIIT
FACULTY OF COMPUTING