

Apeksha Hospital Donor Engagement System

2023-24-100

Status Document 1

Punchihewa S.N

BSc (Hons) in Information Technology specializing in Software Engineering

Department of Information Technology

Sri Lanka Institute of Information Technology Sri Lanka

August 2023

Intelligent Donor-Driven Inventory System for Essential Items

2023-24-100

Status Document 1

Punchihewa S.N

BSc (Hons) in Information Technology specializing in Software Engineering

Department of Information Technology

Sri Lanka Institute of Information Technology Sri Lanka

August 2023

Declaration of The Candidate & Supervisor

I declare that this is my work, and this proposal does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any other university or institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgment is made in the text.

Name	Student ID	Signature
Punchihewa S. N	IT20665166	

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

Signature of the supervisor:

Date:

Signature of the Co-supervisor:

Date:

Table of Contents

1.	Introduction	IV
2.	Gantt Chart	V
3.	Work-Break-Down Chart.....	VI
4.	Screenshot of Meetings and Conversations.....	VII
5.	Project Management Tools & Screenshot.....	XI
5.1	Overall Backlog.....	XI
5.2	Personal Backlog.....	XII
6.	Trained Model Screenshot.....	XIII
6.1	Linear Regression.....	XIII
6.2	Decision Tree Regression.....	XVII

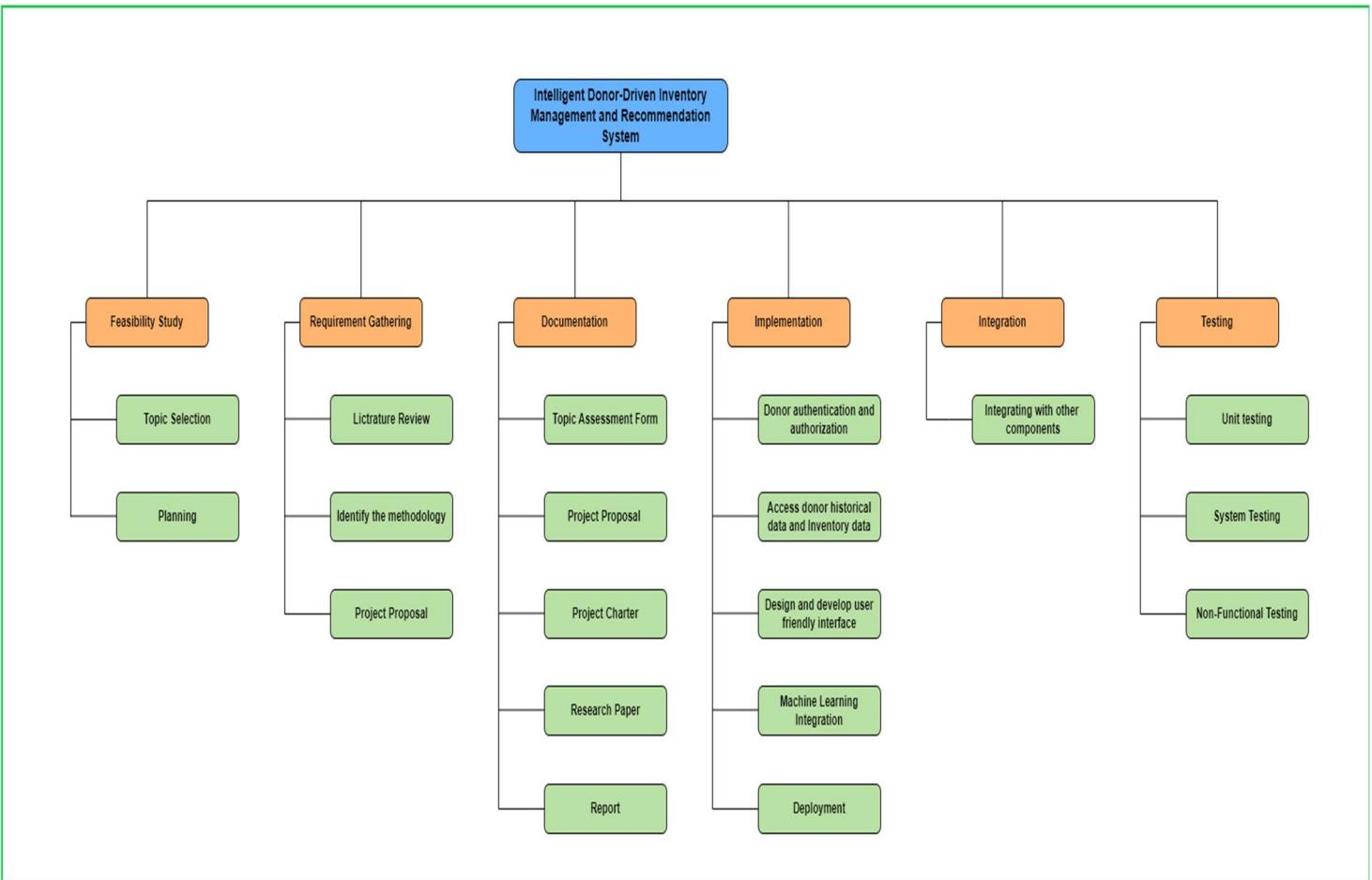
1. Introduction

Apeksha Hospital has the significant problem of enhancing donor involvement and enhancing the management of essential item inventory. To overcome this difficulty, this study proposes a new Intelligent Donor-Driven Inventory System for Essential Supplies. The main goal is to make use of machine learning algorithms to better understand donor habits, preferences, and giving patterns over time. As a result of this knowledge, the system will be in a position to skillfully suggest essential products to prospective donors, therefore stimulating their donations that are in line with the hospital's immediate demands. The ultimate goal of this research is to provide an engaging platform that integrates donor engagement with inventory control. Donors will be able to see the direct results of their donations and get personalized product recommendations from the system. By using these strategies, we want to increase donor pleasure and engagement while establishing a positive cycle of engagement. The research being conducted hopes to make the present of essential items more useful by concluding the gap between what donors want to do and what hospitals need. This collaborative method, which is made possible by machine learning and a donor-centered interface, not only improves the way resources are used but also makes the experience for donors as a whole better. Because of this, the results of this research are likely to make a big difference in how people give donations to Apeksha Hospital.

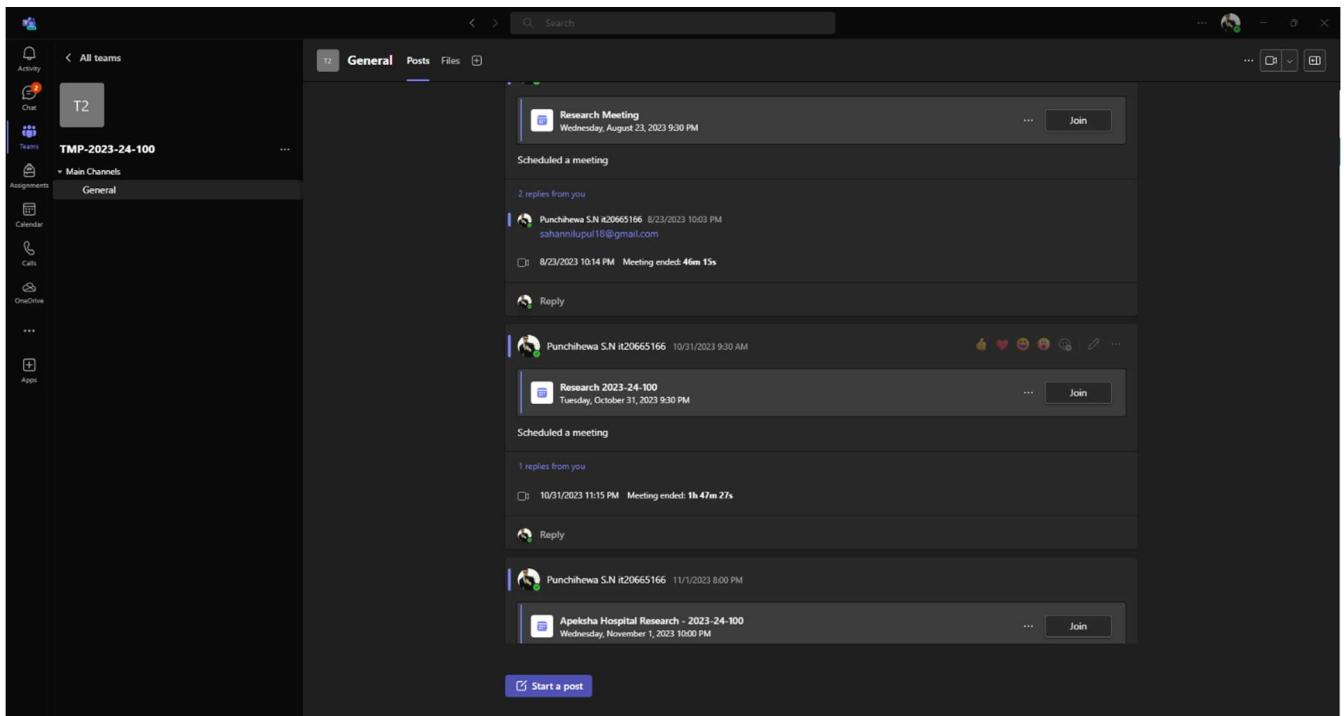
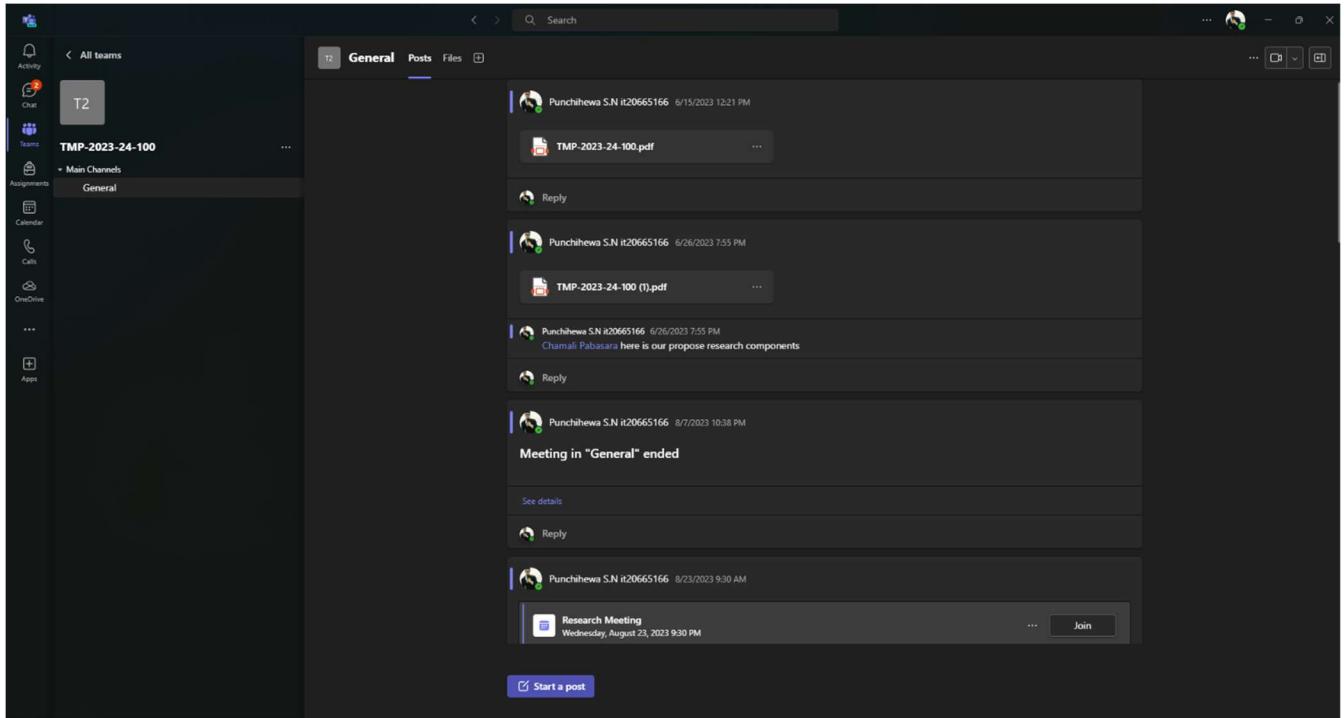
2. Gantt Chart

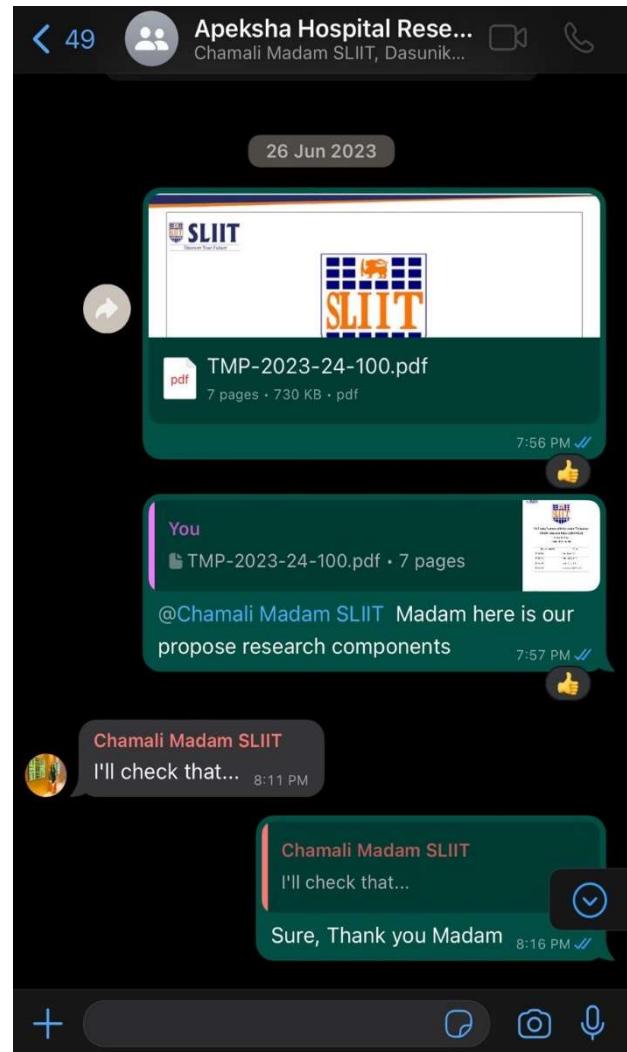
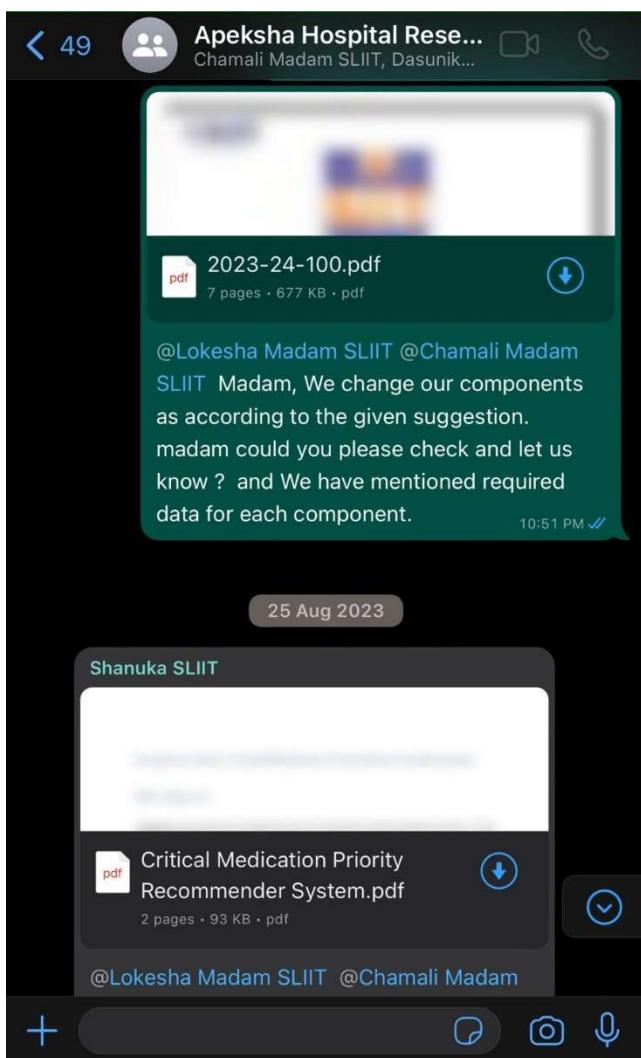
Task Name	Semester 1						Semester 2					
	June	July	August	September	October	November	December	January	February	March	April	May
Feasibility Study	Red											
Topic Selection	Yellow											
Topic Evaluation	Yellow											
Background Study	Red	Red	Yellow	Yellow								
Background Study and Literature Survey												
Proposal Draft												
Project Proposal		Yellow	Yellow	Yellow								
Proposal Presentation												
Proposal Report												
Project Initiation			Red	Red	Yellow	Yellow						
Data gathering				Yellow	Yellow	Yellow						
Selecting technologies												
Project Implementation					Red	Red	Red					
Donor authentication and authorization					Yellow	Yellow	Yellow	Yellow	Yellow			
Access donor historical data and inventory data					Yellow	Yellow	Yellow	Yellow	Yellow			
Design and Develop a user friendly interface					Yellow	Yellow	Yellow	Yellow	Yellow			
Machine Learning Integration					Yellow	Yellow	Yellow	Yellow	Yellow			
Research Paper					Yellow	Yellow	Yellow	Yellow	Yellow			
Final Implementation					Yellow	Yellow	Yellow	Yellow	Yellow			
Unit Testing					Yellow	Yellow	Yellow	Yellow	Yellow			
Evaluating and error fixing					Yellow	Yellow	Yellow	Yellow	Yellow			
Integrating with other components					Yellow	Yellow	Yellow	Yellow	Yellow			
Final Stages						Red	Red	Red	Red			
System Integration						Yellow	Yellow	Yellow	Yellow			
System Testing						Yellow	Yellow	Yellow	Yellow			
Evaluating and error fixing						Yellow	Yellow	Yellow	Yellow			
Final Report						Yellow	Yellow	Yellow	Yellow			

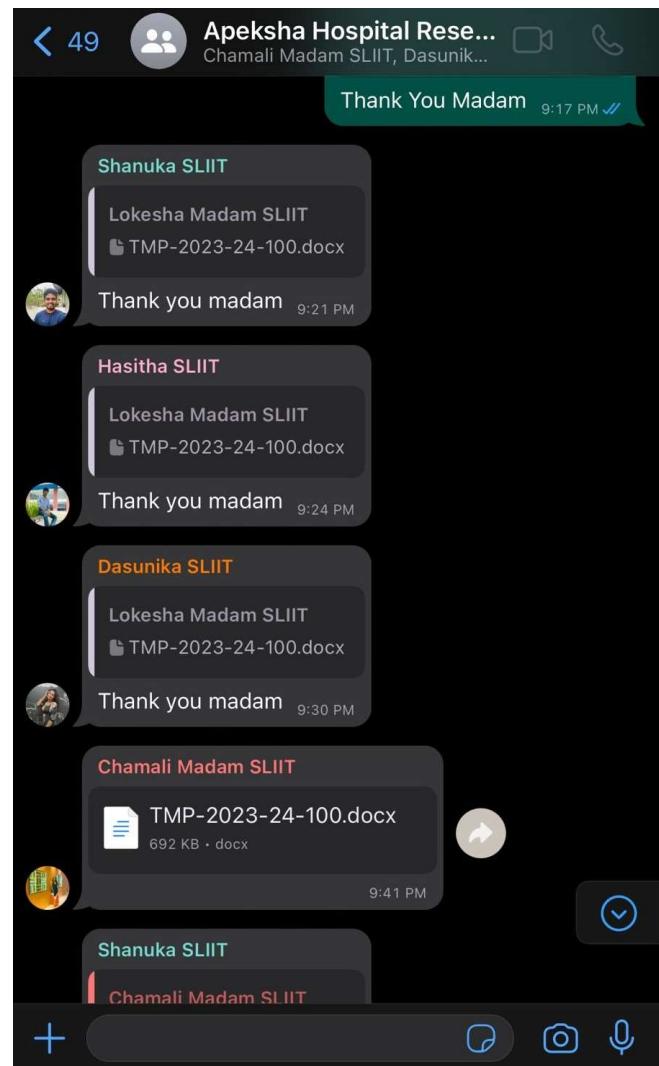
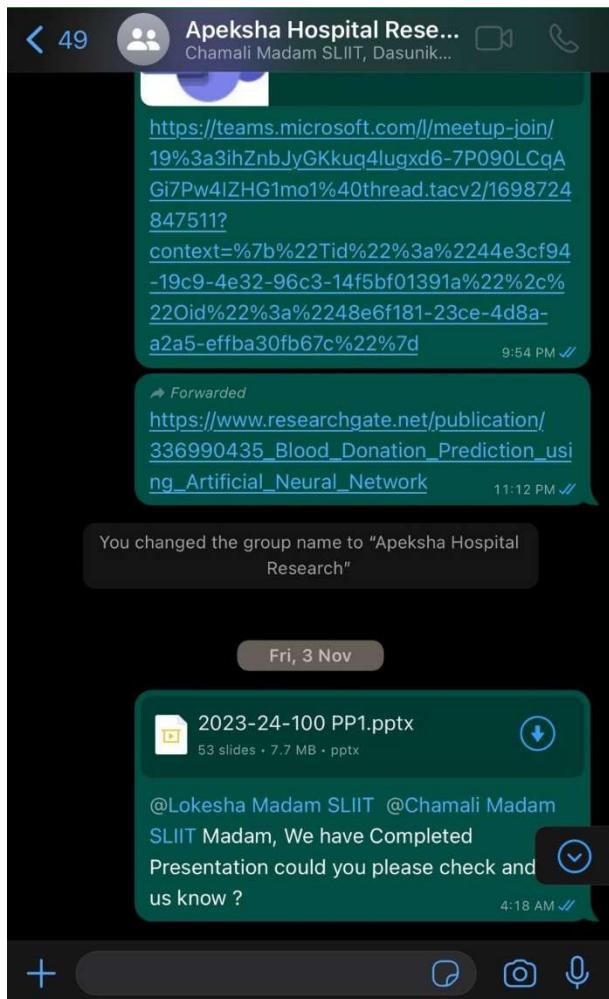
3. Work-Break-Down Chart

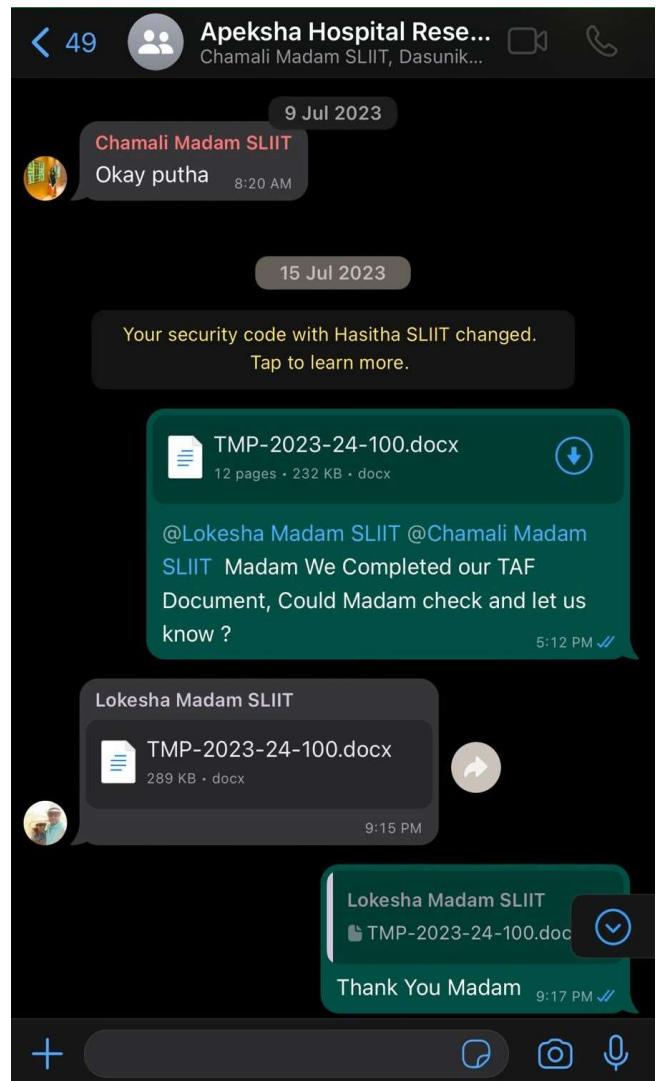
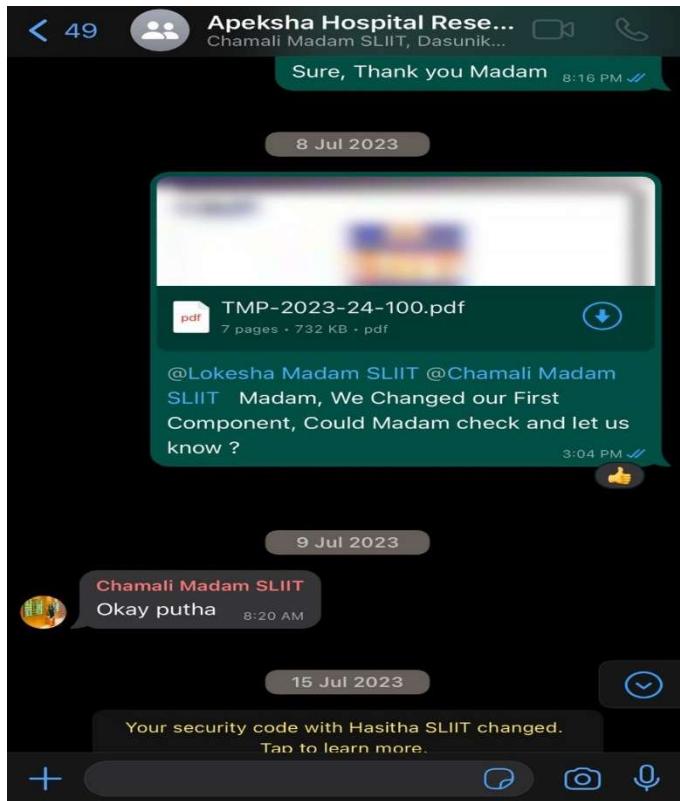
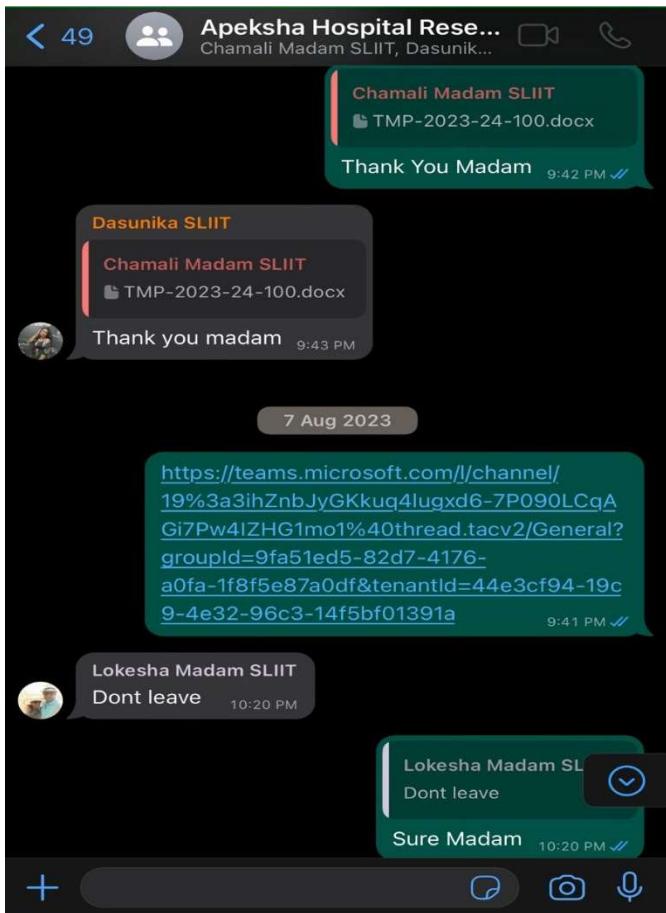


4. Screenshot of Meetings and Conversations









X

5. Project Management Tools & Screenshot

5.1 Overall Backlog

The screenshot shows the Azure DevOps Boards backlog for the "Apexka Hospital Donor Engag..." project. The backlog is organized into four columns: New, Active, Resolved, and Closed. The New column contains five items, the Active column contains 15 items, the Resolved column contains 16 items, and the Closed column contains 13 items. Each item card includes a summary, status (Active or Resolved), and assigned developer.

Column	Count	Items
New	5	44 Combine all the models to implement the final model 29 Implement the Medication Dashboard User Interface. 30 Implement items prediction dashboard 32 Implement the Prediction Dashboard User Interface.
Active	15	46 Model Implemented to identify Dryness of Hair 48 Implement the Web Application with User-friendly Interface 47 Model Implemented to identify bleached of Hair 14 Collecting Real Medication Data from Apexka Hospital 18 Collecting Real Donation Campign historical data from Apelsha Hospital 17 Collecting Real Data from Apelsha Hospital for Intelligent Donor-Driven Inventory System for Esesential Items
Resolved	16	53 Model Implemented to identify Dandruff & lice of Hair 52 Model Implemented to identify length type of Hair 51 Model Implemented to identify Color of Hair 50 Data Pre-Processing and Create Data Set With Data Augmentation 19 Creating test data set to feed the model 49 Identification of best architecture for transfer learning
Closed	13	45 Collection Test-Data- Promoting Quality Hair Donation for Cancer Patients 15 Collecting Test Data - Previous Donation campaign attendees count and date 16 Collection Test Data - Intelligent Donor-Driven Inventory System for Essential Items 13 Collecting Test Data-Critical Medication Identification and shortage

5.2 Personal Backlog

The screenshot shows the Azure DevOps Boards backlog view for the project "Apeksha Hospital Donor Engagement System-2023-24-100 Team". The backlog is organized into four columns: New, Active, Resolved, and Closed. The "New" column contains one item: "30 Implement items prediction dashboard" (New, Active). The "Active" column contains three items: "17 Collecting Real Data from Apeksha Hospital for Intelligent Donor-Driven Inventory System for Essential Items" (New, Active), "21 Creating Test data for feed model" (Resolved), and "22 Pre-Process data set" (Resolved). The "Resolved" column contains two items: "25 Implement Donor-Driven Inventory System for Essential Items Model" (Resolved) and "27 Identify suitable algorithms for train machine learning model" (Resolved). The "Closed" column contains one item: "16 Collection Test Data - Intelligent Donor-Driven Inventory System for Essential Items" (Closed). The left sidebar shows navigation links for Overview, Boards, Work items, Boards, Backlogs, Sprints, Queries, Delivery Plans, Analytics views, Repos, Pipelines, Test Plans, and Artifacts. The bottom left corner shows "Project settings".

New	Active	Resolved	Closed
30 Implement items prediction dashboard • New Punchihewa S.N it20665166	17 Collecting Real Data from Apeksha Hospital for Intelligent Donor-Driven Inventory System for Essential Items • New Punchihewa S.N it20665166 21 Creating Test data for feed model • Resolved Punchihewa S.N it20665166 22 Pre-Process data set • Resolved Punchihewa S.N it20665166	25 Implement Donor-Driven Inventory System for Essential Items Model • Resolved Punchihewa S.N it20665166 27 Identify suitable algorithms for train machine learning model • Resolved Punchihewa S.N it20665166	16 Collection Test Data - Intelligent Donor-Driven Inventory System for Essential Items • Closed Punchihewa S.N it20665166

6. Trained Model Screenshot

6.1 Linear Regression

The screenshot shows a Jupyter Notebook interface with the title "jupyter InventoryPrediction Last Checkpoint: 4 hours ago (autosaved)". The toolbar includes File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Trusted, and Python 3 (ipykernel). Below the toolbar are standard notebook controls for cell selection, running, and saving.

The notebook contains the following code:

```
In [1]: import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import accuracy_score, classification_report
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import ConfusionMatrixDisplay
import matplotlib.pyplot as plt

In [2]: # Load the dataset
df = pd.read_csv('newdataset.csv')

In [3]: # calculate the correlation coefficient between the "RequestedQuantity" column in the inventory DataFrame and all the other numeric columns
df.corr(numeric_only = ['RequestedQuantity'])["RequestedQuantity"]

Out[3]: ItemID      -0.174915
ItemCategory   -0.210888
QuantityInStock  0.244093
UsageHistory    -0.522111
RequestedQuantity 1.000000
Demand          0.044951
Name: RequestedQuantity, dtype: float64

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

In [5]: # Define independent variables (features) and the target variable
X = df[['ItemName", "ItemID", "ItemCategory", "UsageHistory"]]
y = df["RequestedQuantity"]

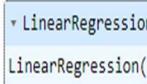
In [6]: # Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=10)

In [7]: # print out the shape of the training and testing sets
print("Training set shape : ", X_train.shape, y_train.shape)
print("Testing set shape : ", X_test.shape, y_test.shape)
```

jupyter InventoryPrediction Last Checkpoint: 4 hours ago (autosaved)  Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel) O

In [8]: # Initialize linear regression model Calculate accuracy score trained Linear regression model
model = LinearRegression()
model.fit(X_train, y_train)

Out[8]:  LinearRegression
LinearRegression()

In [9]: # Make predictions on the test set
y_pred = model.predict(X_test)

In [10]: # Test Set prediction
data = {
 "ItemName": "FaceMask",
 "ItemID": 100,
 "ItemCategory": 1,
 "UsageHistory": 45,
}

Convert gender to numerical format
if "ItemName" in data:
 data["ItemName"] = le_ItemName.transform([data["ItemName"]])[0]

Ensure the data is in the correct format
input_data = pd.DataFrame(
 [data],
 columns=[
 "ItemName",
 "ItemID",
 "ItemCategory",
 "UsageHistory",
],
)

In [11]: # Make a prediction
prediction = model.predict(input_data).tolist()

In [12]: # Predict 'RequestedQuantity' on the test set
prediction

```
In [11]: # Make a prediction  
prediction = model.predict(input_data).tolist()
```

```
In [12]: # Predict 'RequestedQuantity' on the test set  
prediction
```

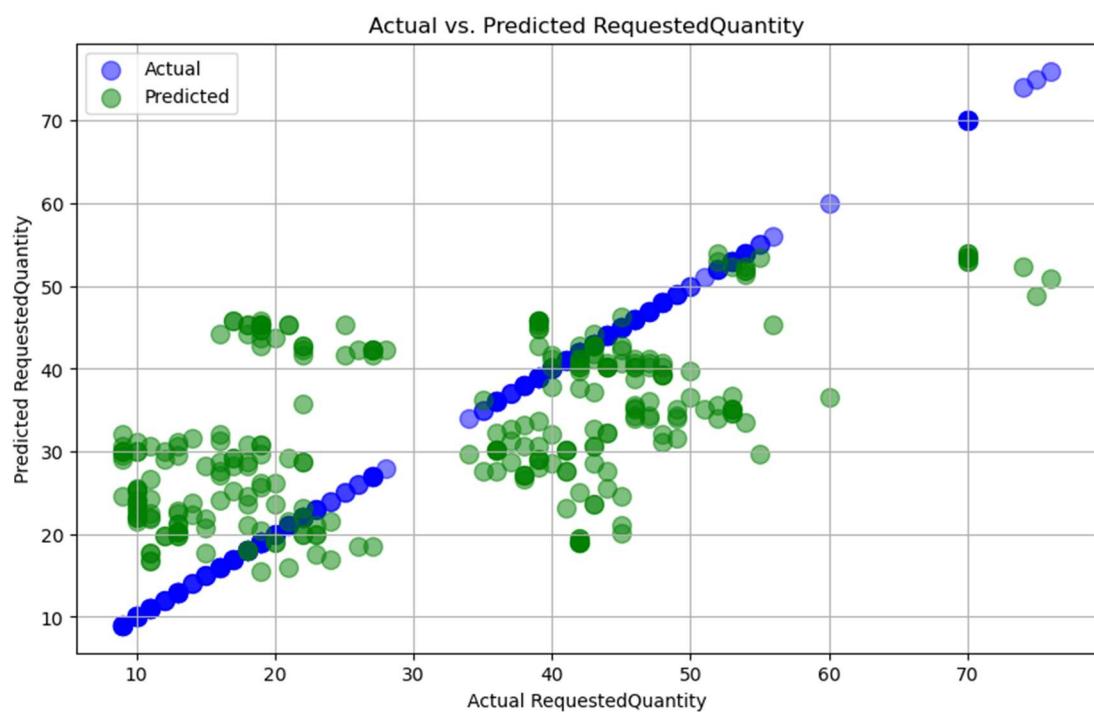
```
Out[12]: [37.79863989744041]
```

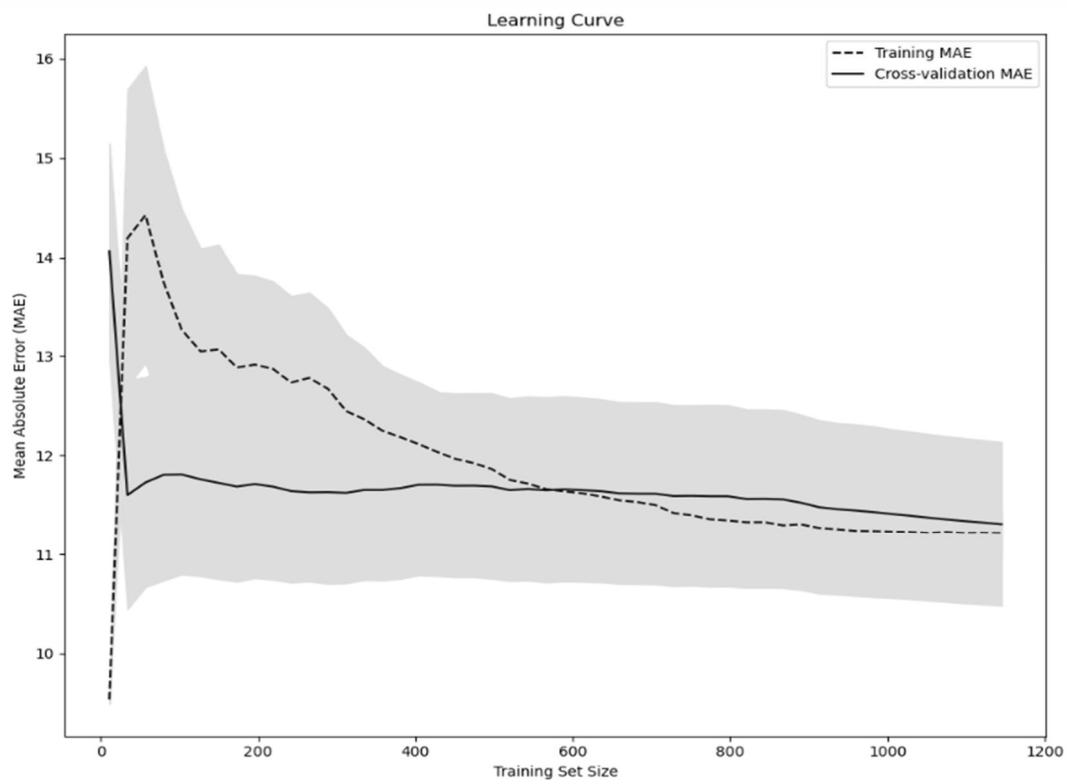
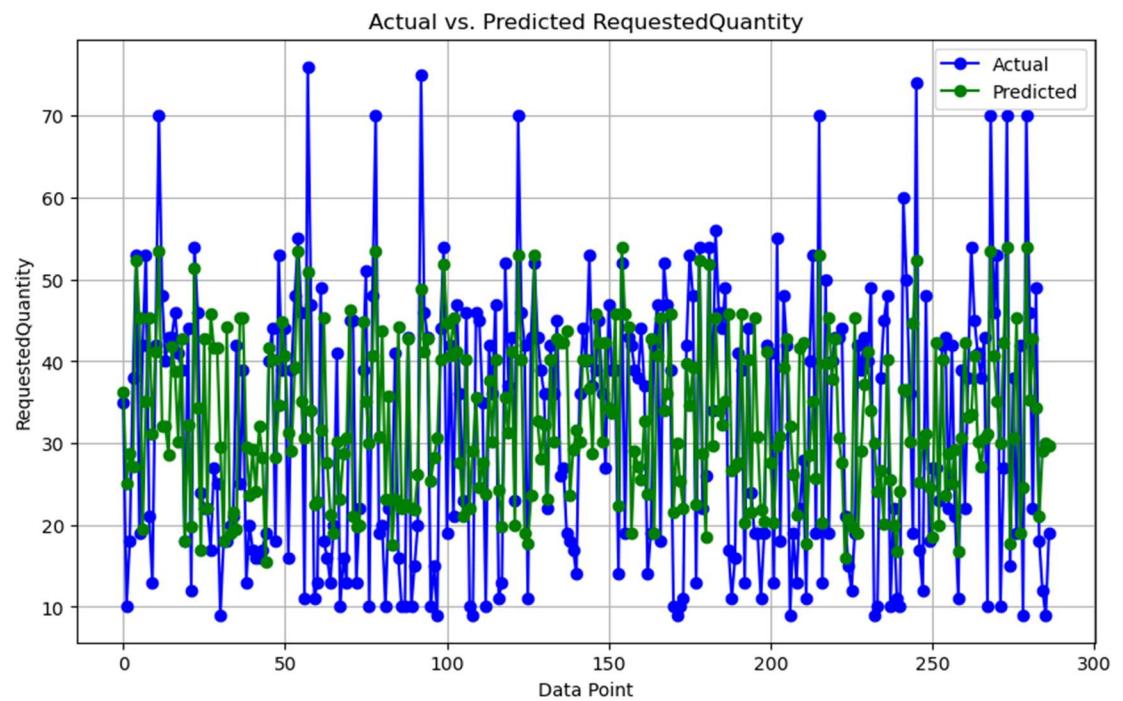
```
In [13]: prediction
```

```
Out[13]: [37.79863989744041]
```

```
In [14]: # model.fit(X_train, y_train)  
model.score(X_train, y_train)
```

```
Out[14]: 0.36888600431470053
```





6.2 Decision Tree Regression

Training Set Size

```
In [20]: # Decision Tree Regression Algorithm

In [21]: import pandas as pd
         from sklearn.model_selection import train_test_split
         from sklearn.tree import DecisionTreeRegressor # Import DecisionTreeRegressor
         import matplotlib.pyplot as plt

In [22]: # Load the dataset
         df = pd.read_csv('newdataset.csv')

         # Preprocess the Data
         # (You can keep the LabelEncoder for ItemName)
         from sklearn.preprocessing import LabelEncoder

In [23]: le_ItemName = LabelEncoder()
         df["ItemName"] = le_ItemName.fit_transform(df["ItemName"])

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

In [24]: # 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)

In [25]: # Print out the shape of the training and testing sets
         print("Training set shape:", X_train.shape, y_train.shape)
         print("Testing set shape:", X_test.shape, y_test.shape)

         Training set shape: (1145, 4) (1145,)
         Testing set shape: (287, 4) (287,)

In [26]: # Initialize Decision Tree Regression model
         model = DecisionTreeRegressor(random_state=42) # You can specify other hyperparameters if needed

In [27]: # Train the Decision Tree model
         model.fit(X_train, y_train)

Out[27]: - DecisionTreeRegressor
          DecisionTreeRegressor(random_state=42)
```

```
[DECISSIONTREEREGRESSOR(random_state=42)]
```

```
In [28]: # Test Set prediction
data = {
    "ItemName": "FaceMask",
    "ItemID": 100,
    "ItemCategory": 1,
    "UsageHistory": 45,
}

# Convert "ItemName" to numerical format using the LabelEncoder
if "ItemName" in data:
    data["ItemName"] = le_ItemName.transform([data["ItemName"]])[0]

# Ensure the data is in the correct format
input_data = pd.DataFrame(
    [data],
    columns=[
        "ItemName",
        "ItemID",
        "ItemCategory",
        "UsageHistory",
    ],
)
```

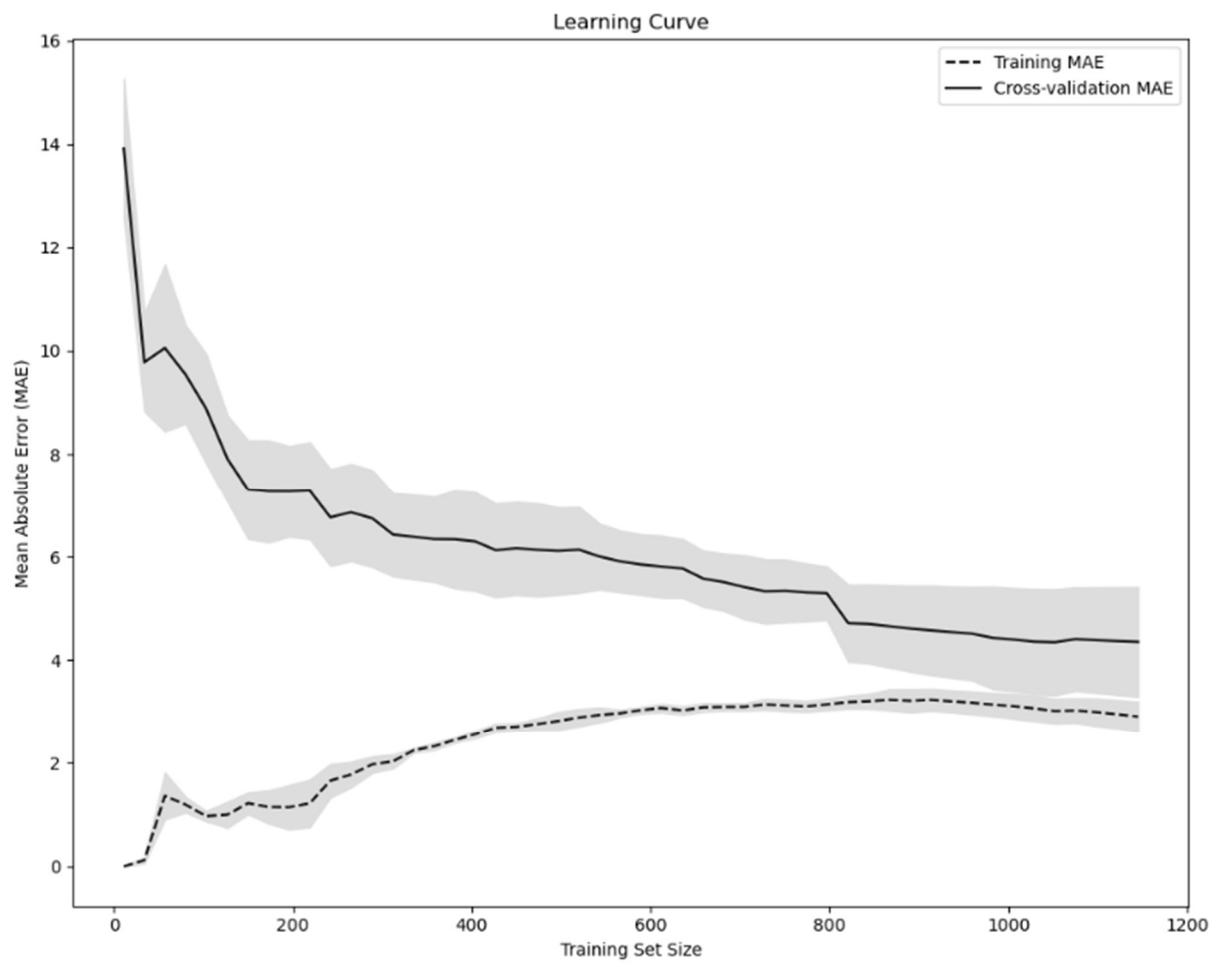
```
In [29]: # Make a prediction using the Decision Tree model
prediction = model.predict(input_data).tolist()

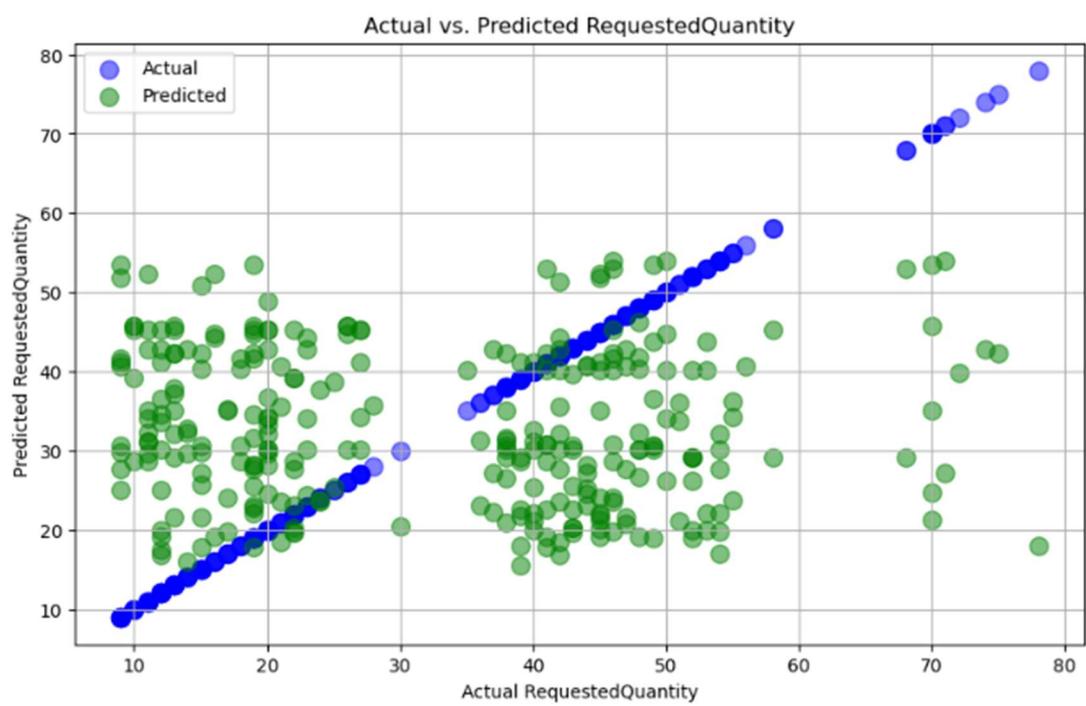
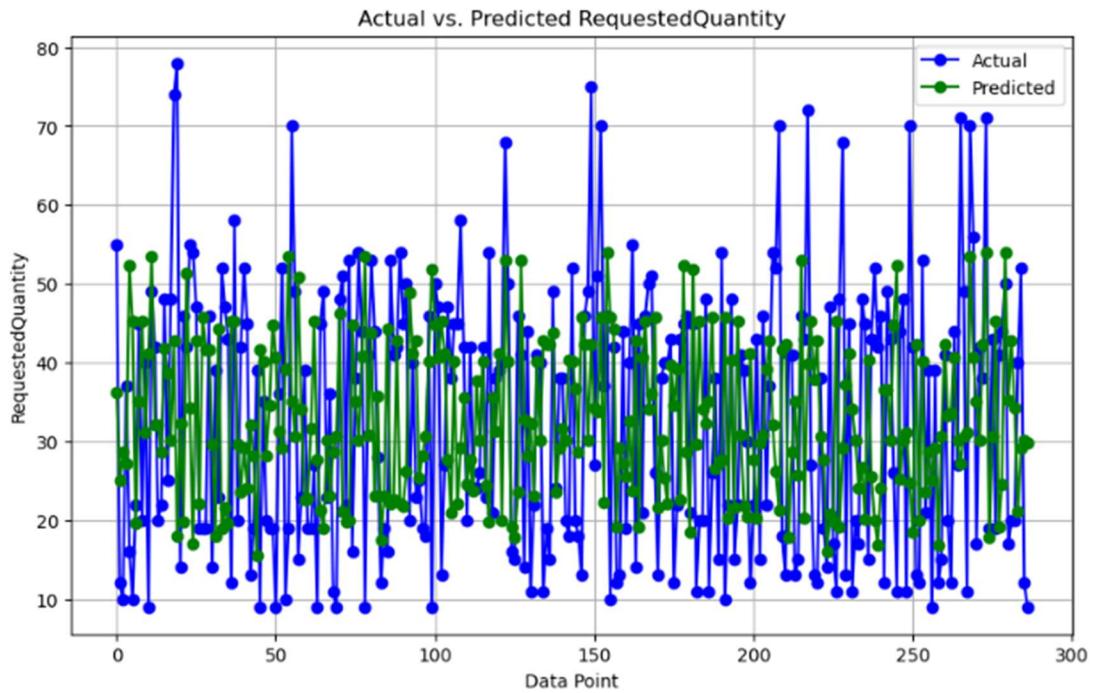
# Print the predicted 'RequestedQuantity' on the test set
print("Prediction:", prediction)

Prediction: [46.66666666666664]
```

```
In [30]: model.score(X_train, y_train)

out[30]: 0.8945812178633263
```





XX

Apeksha Hospital Donor Engagement System

2023-24-100

Status Document 01

Prabodha K.W.D.S

BSc (Hons) in Information Technology specializing in Software Engineering

Department of Information Technology

Sri Lanka Institute of Information Technology Sri Lanka

August 2023

Critical Medication Priority Recommender System

2023-24-100

Status Document 01

Prabodha K.W.D.S

BSc (Hons) in Information Technology specializing in Software Engineering

Department of Information Technology

Sri Lanka Institute of Information Technology Sri Lanka

August 2023

Declaration of The Candidate & Supervisor

I declare that this is my work, and this proposal does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any other university or institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgment is made in the text.

Name	Student ID	Signature
Prabodha K.W.D. S	IT20665098	

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

Signature of the supervisor:

Date:

Signature of the Co-supervisor:

Date:

Table of Contents

Introduction	IV
Work Break-Down Structure.....	VI
Screenshots of meetings and conversations with supervisor	VII
Personal Backlog	IX
Overall Backlog.....	IX
Implemented Machine Learning Models.....	X
Critical Medication Identification Model	X
Shortage Prediction Model	X

Introduction

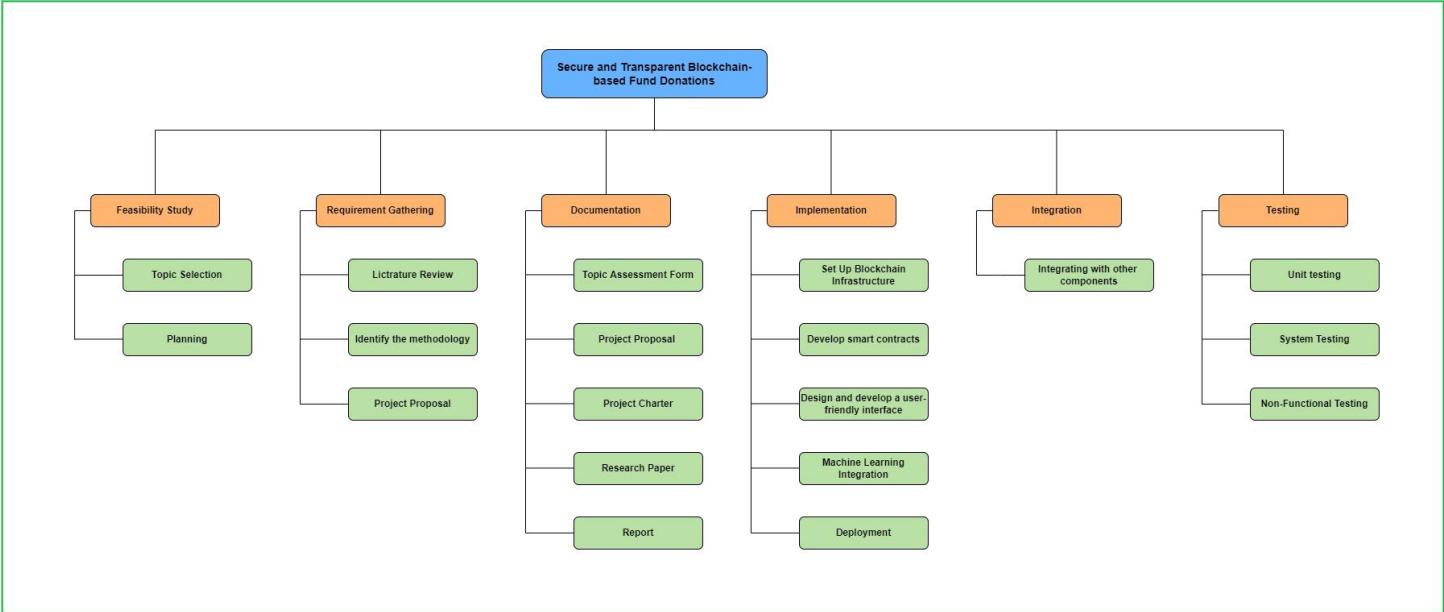
The "Critical Medication Priority Recommender System" presents a novel machine learning-based solution aimed at addressing the persistent challenge faced by Apeksha Hospital in identifying and prioritizing essential medications in low supply, ultimately affecting patient care quality. This research component outlines the development of a medication recommendation platform that strategically assesses historical medication usage, donation records, and medication attributes. The system integrates two pivotal machine learning models: a "Shortage Prediction" model to anticipate medication scarcities, a "Critical Medication Identification" model ranking medications based on medical significance. This process is further enhanced through optimization algorithms that rigorously prioritize medications by importance, availability, and expiration. Leveraging existing medication donation frameworks, this research not only fills a critical gap in medication allocation but also bolsters resource efficiency, offering a comprehensive solution for ensuring high-quality patient care in the face of limited resources.

Keywords: Medication shortage, Machine learning-based system, Shortage prediction model, Donation utilization

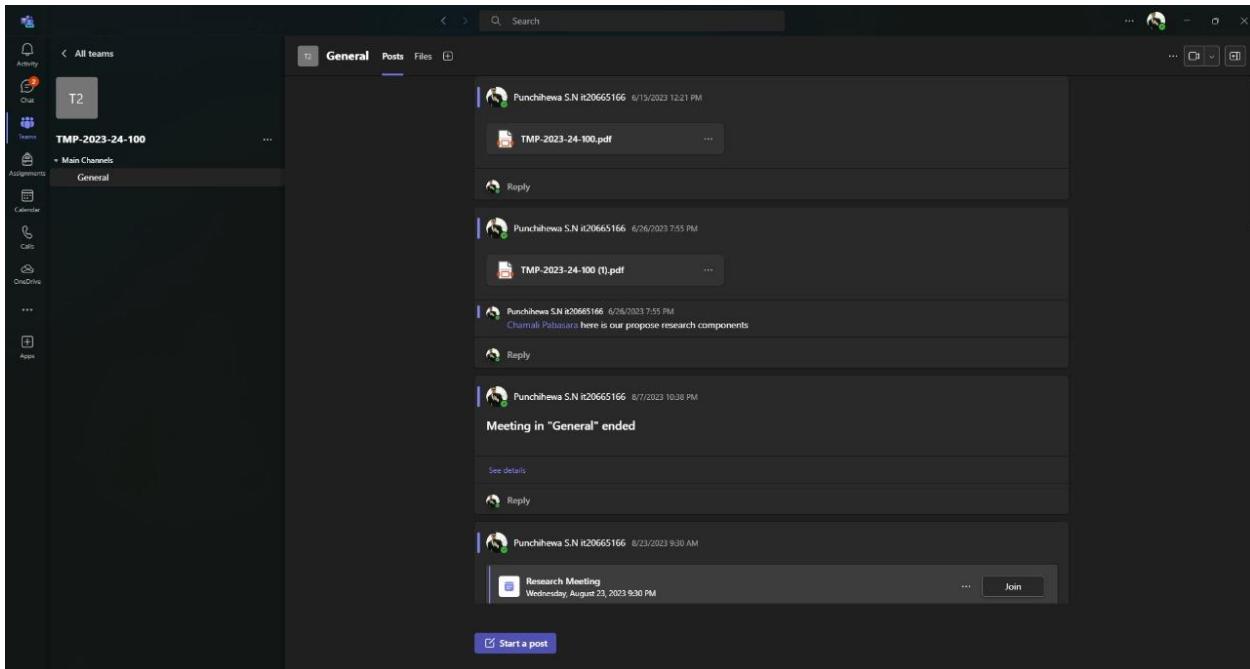
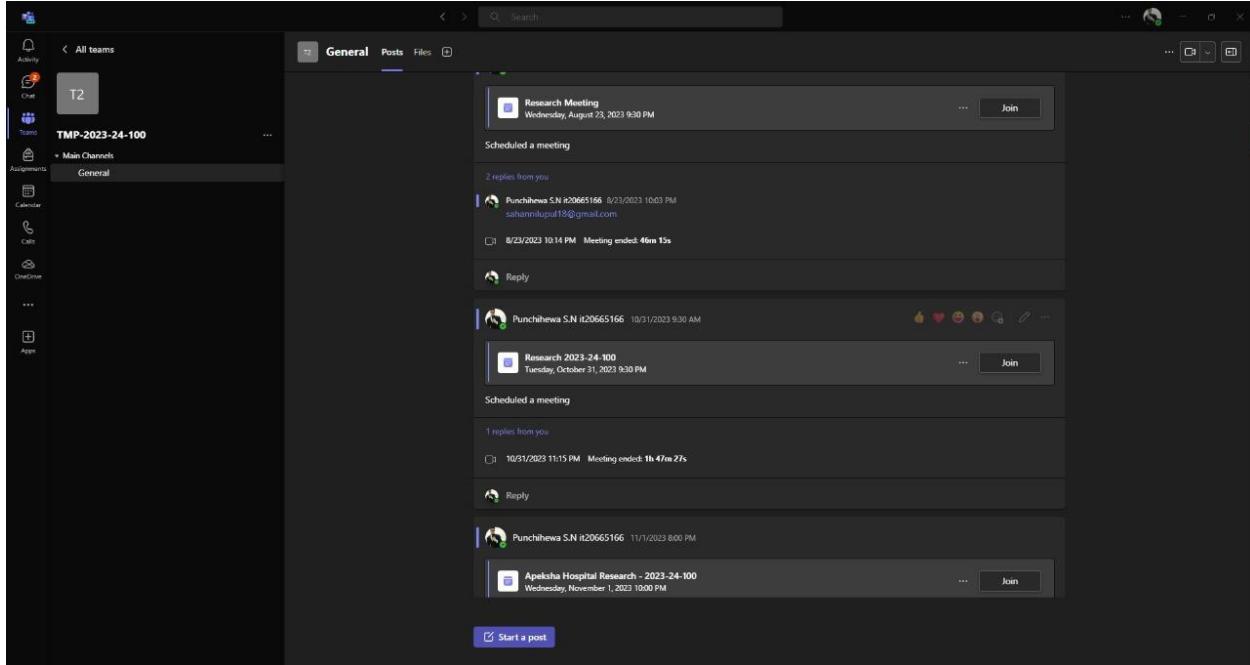
Gantt Chart

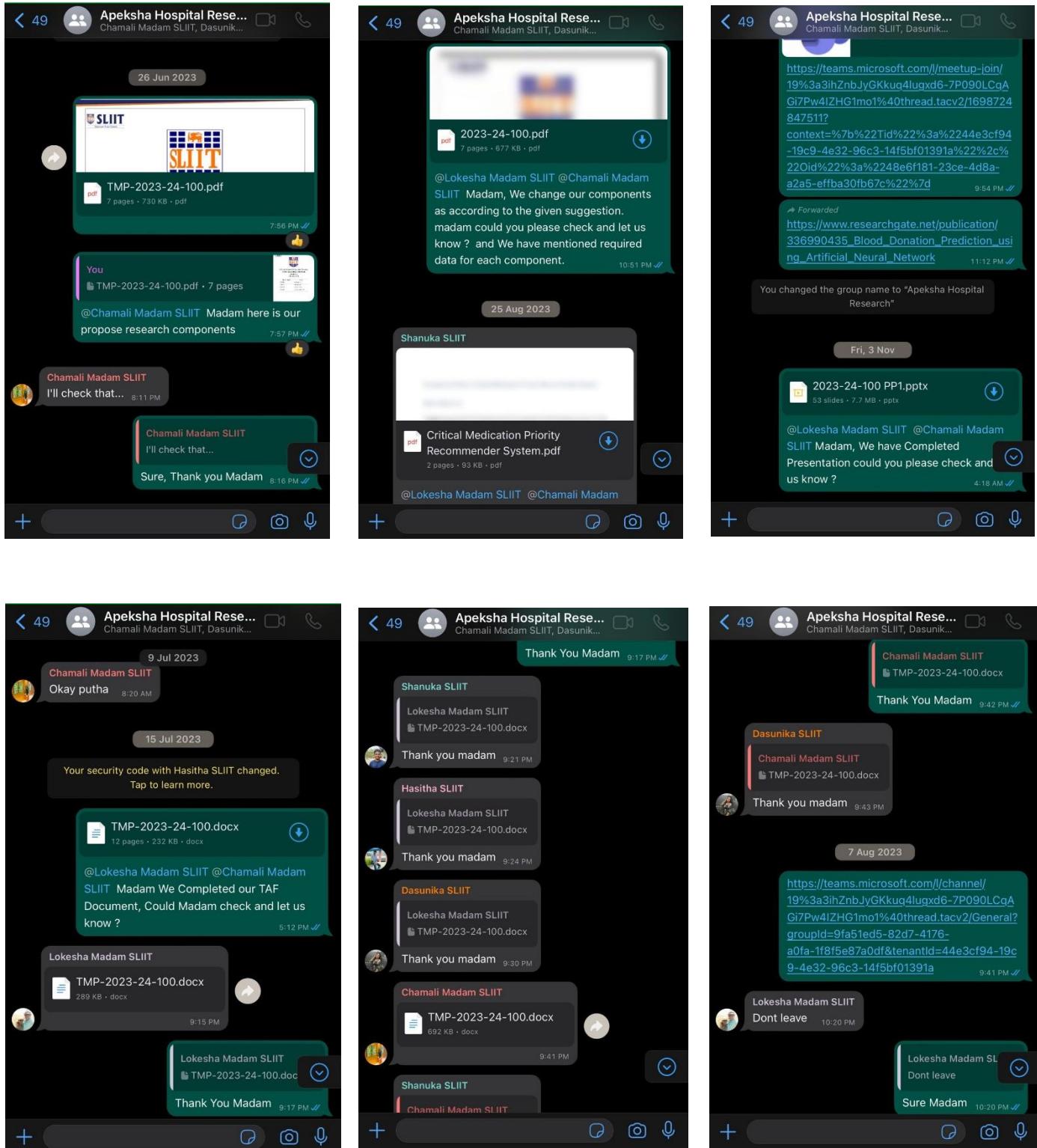
Task Name	Semester 1						Semester 2					
	June	July	August	September	October	November	December	January	February	March	April	May
Feasibility Study	Red											
Topic Selection	Yellow	Yellow										
Topic Evaluation												
Background Study	Red	Red	Yellow									
Background Study and Literature Survey												
Proposal Draft												
Project Proposal		Red	Yellow									
Proposal Presentation												
Proposal Report												
Project Initiation			Red	Yellow								
Data gathering				Yellow	Yellow							
Selecting technologies												
Project Implementation				Yellow	Yellow	Red	Red	Red	Red	Red	Red	Red
Set up Blockchain Infrastructure												
Develop smart Contracts												
Design and Develop a user friendly interface												
Machine Learning Integration												
Research Paper												
Final Implementation												
Unit Testing												
Evaluating and error fixing												
Integrating with other components												
Final Stages								Red	Red	Red	Red	Red
System Integration								Yellow	Yellow	Yellow	Yellow	Yellow
System Testing												
Evaluating and error fixing												
Final Report										Yellow	Yellow	Yellow

Work Break-Down Structure



Screenshots of meetings and conversations with supervisor





Personal Backlog

The screenshot shows a 'Personal Backlog' board for the 'Apeksha Hospital Donor Engagement' project. The board has four columns: 'New', 'Active', 'Resolved', and 'Closed'. The 'New' column contains one item: '29 Implement the Medication Dashboard User Interface.' The 'Active' column contains five items: '14 Collecting Real Medication Data from Apeksha Hospital', '19 Creating test data set to feed the model', '20 Pre-process the Data set', '28 Implementing the Critical Medication Shortage Prediction model using Logistic Regression Algorithm.', and '24 Implementing the Critical Medication Identification model using Logistic Regression Algorithm.'. The 'Resolved' column contains three items: '13 Collecting Test Data-Critical Medication Identification and shortage', '19 Creating test data set to feed the model', and '24 Implementing the Critical Medication Identification model using Logistic Regression Algorithm.'. The 'Closed' column contains one item: '13 Collecting Test Data-Critical Medication Identification and shortage'. The board includes filters for 'Types', 'Shanuka Praboda', 'Tags', 'Iteration', 'Area', and 'Parent Work Item'.

Overall Backlog

The screenshot shows the 'Overall Backlog' board for the 'Apeksha Hospital Donor Engagement System-2023-24-100 Team'. The board has four columns: 'New', 'Active', 'Resolved', and 'Closed'. The 'New' column contains five items: '44 Combine all the models to implement the final model', '29 Implement the Medication Dashboard User Interface.', '30 Implement items prediction dashboard', '32 Implement the Prediction Dashboard User Interface.', and '33 Implement the Medication Dashboard User Interface.'. The 'Active' column contains ten items: '46 Model Implemented to identify Dryness of Hair', '47 Model Implemented to identify bleached of Hair', '48 Implement the Web Application with User-friendly Interface', '49 Model Implemented to identify length type of Hair', '50 Data Pre-Processing and Create Data Set With Data Augmentation', '51 Model Implemented to identify Color of Hair', '52 Model Implemented to identify Dandruff & lice of Hair', '53 Model Implemented to identify Dandruff & lice of Hair', '54 Collection Test Data - Promoting Quality Hair Donation for Cancer Patients', and '55 Collection Test Data - Previous Donation campaign attendees count and date'. The 'Resolved' column contains six items: '45 Collection Test Data - Promoting Quality Hair Donation for Cancer Patients', '46 Model Implemented to identify Dryness of Hair', '47 Model Implemented to identify bleached of Hair', '48 Implement the Web Application with User-friendly Interface', '49 Model Implemented to identify length type of Hair', and '50 Data Pre-Processing and Create Data Set With Data Augmentation'. The 'Closed' column contains three items: '16 Collection Test Data - Intelligent Donor-Driven Inventory System for Essential Items', '17 Collecting Real Data from Apeksha Hospital for Intelligent Donor-Driven Inventory System for Essential Items', and '18 Collecting Real Donation Campaign historical data from Apeksha Hospital'. The board includes filters for 'Features' and other project settings.

Implemented Machine Learning Models

Critical Medication Identification Model

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
from sklearn.linear_model import LogisticRegression

# Read the CSV file into a DataFrame
data = pd.read_csv('critical_medication.csv')

data['Shelf_Life'] = pd.to_datetime(data['Shelf_Life'])
data['Shelf_Life_Day'] = data['Shelf_Life'].dt.day
data['Shelf_Life_Month'] = data['Shelf_Life'].dt.month
data['Shelf_Life_Year'] = data['Shelf_Life'].dt.year

data.drop(['Shelf_Life'], axis=1, inplace=True)

# Define the features (predictors) and the target variable
X = data[['Patients', 'Usage', 'Emergency_Usage', 'Shelf_Life_Day', 'Shelf_Life_Month', 'Shelf_Life_Year']]
y = data['Importance_Level']

# 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)

log_reg=LogisticRegression(random_state=0).fit(X_train,y_train)

log_reg.predict(X_train)
```

Shortage Prediction Model

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression

df=pd.read_csv("shortage_data.csv")

df['Shelf_Life'] = pd.to_datetime(df['Shelf_Life'])
df['Admin_Timestamp'] = pd.to_datetime(df['Admin_Timestamp'])

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

df.drop(['Shelf_Life', 'Admin_Timestamp'], axis=1, inplace=True)
X=df.drop(columns='Medication Shortage')

y=df['Medication Shortage']
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=21)
X_train

scaler=StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled=scaler.transform(X_test)
X_train_scaled

log_reg=LogisticRegression(random_state=0).fit(X_train_scaled,y_train)
log_reg.predict(X_train_scaled)
log_reg.score(X_train_scaled, y_train)
```

Apeksha Hospital Donor Engagement System

2023-24-100

Status Document 1

Bandara H.R.H.S

BSc (Hons) in Information Technology specializing in Software Engineering

Department of Information Technology

Sri Lanka Institute of Information Technology Sri Lanka

August 2023

Predictive Analytics for Donation Campaign Success

2023-24-100

Status Document 1

Bandara H.R.H.S

BSc (Hons) in Information Technology specializing in Software Engineering

Department of Information Technology

Sri Lanka Institute of Information Technology Sri Lanka

August 2023

Declaration of The Candidate & Supervisor

I declare that this is my work, and this proposal does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any other university or institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgment is made in the text.

Name	Student ID	Signature
Bandara H.R.H.S	IT20662028	

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

Signature of the supervisor:

Date:

Signature of the Co-supervisor:

Date:

Table of Contents

Introduction.....	IV
Gantt chart.....	V
Work Break down Chart	VI
Screenshots of meetings and conversations with supervisor	VII
Personal backlog	X
Overall Backlog	XI
Trained Model Screenshot.....	XII

Introduction

Blood holds paramount significance for human well-being, particularly among individuals afflicted with blood-related disorders such as leukemia and blood cancer. The pivotal role of blood in medical treatments, especially within Apeksha hospitals, underscores the constant and heightened need for blood donations. However, blood donation campaigns encounter multifaceted challenges impeding their success. Issues like low donor turnout, resource wastage including food, and insufficient medical equipment such as needles, blood collection bags, blood pressure monitors, and donor beds or chairs, along with inadequate medical staff, contribute to the inefficacy of these campaigns.

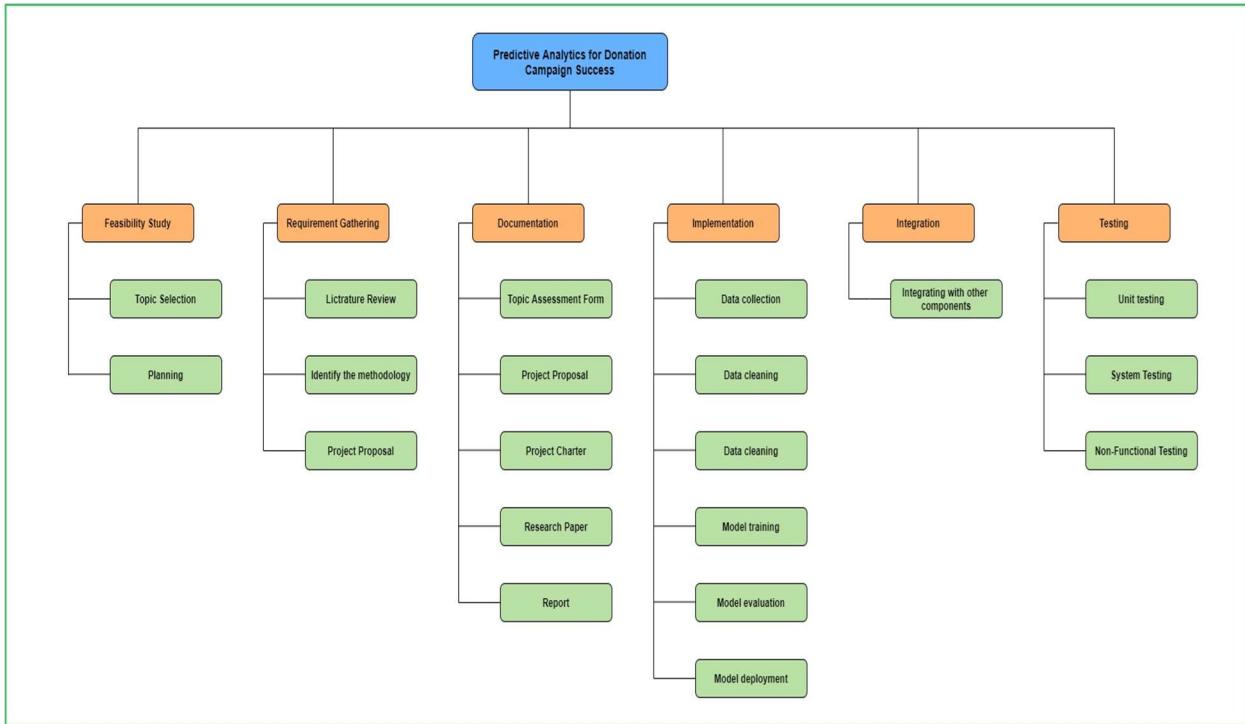
To surmount these obstacles and optimize the allocation of resources, this research advocates for the integration of predictive analytics empowered by machine learning. By harnessing historical data, predictive models can be devised to anticipate the likelihood of campaign success. These models would take into account an array of variables, including donor engagement patterns, external factors, and resource availability. By providing insights into donor behavior trends and optimal campaign conditions, machine learning could profoundly refine the strategic planning and execution of blood donation initiatives, leading to resource optimization. This impact is particularly crucial for Apeksha hospitals, where a constant and heightened supply of blood is indispensable.

Keywords: Machine learning, blood donation campaigns, donor turnout, resource optimization, campaign success, predictive analytics.

Gantt chart

Task Name	Semester 1						Semester 2					
	June	July	August	September	October	November	December	January	February	March	April	May
Feasibility Study	Red											
Topic Selection	Yellow											
Topic Evaluation	Yellow											
Background Study	Red	Red	Yellow									
Background Study and Literature Survey												
Proposal Draft												
Project Proposal			Red	Red								
Proposal Presentation			Yellow	Yellow								
Proposal Report			Yellow	Yellow								
Project Initiation			Red	Red								
Data gathering			Yellow	Yellow								
Selecting technologies			Yellow	Yellow								
Project Implementation			Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow
Set up Blockchain Infrastructure			Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow
Develop smart Contracts			Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow
Design and Develop a user friendly interface			Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow
Machine Learning Integration			Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow
Research Paper			Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow
Final Implementation			Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow
Unit Testing			Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow
Evaluating and error fixing			Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow
Integrating with other components			Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow
Final Stages			Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow
System Integration			Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow
System Testing			Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow
Evaluating and error fixing			Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow
Final Report			Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow	Yellow

Work Break down Chart



Screenshots of meetings and conversations with supervisor

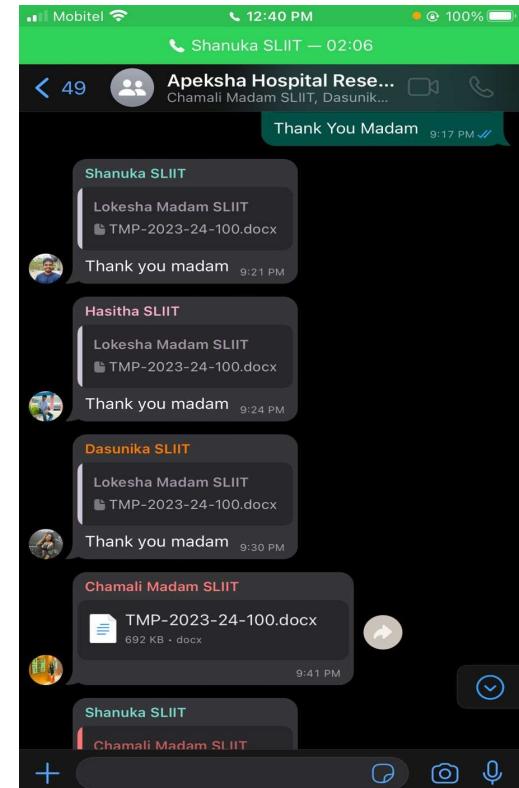
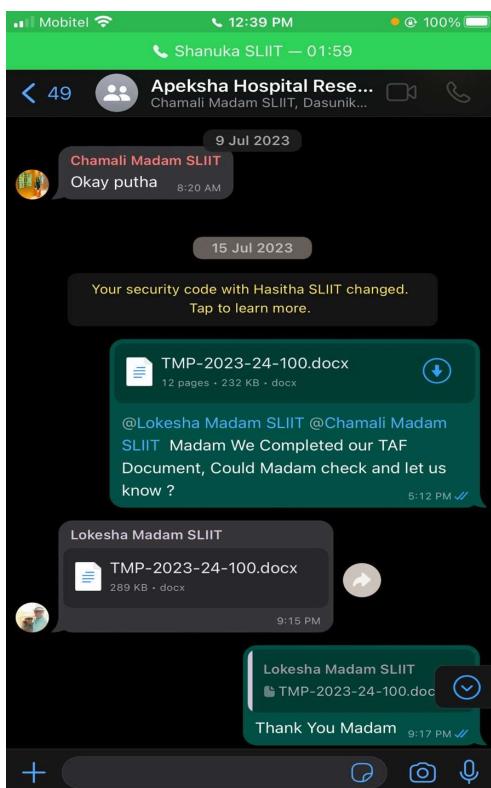
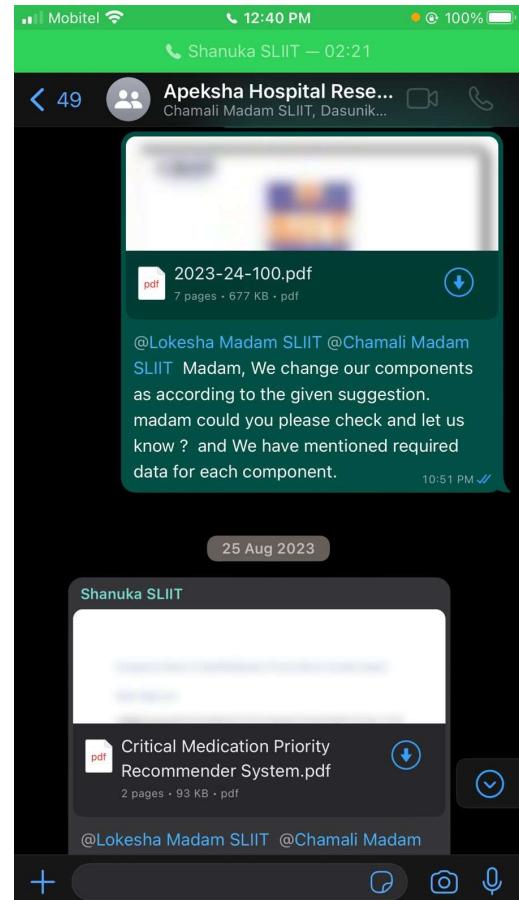
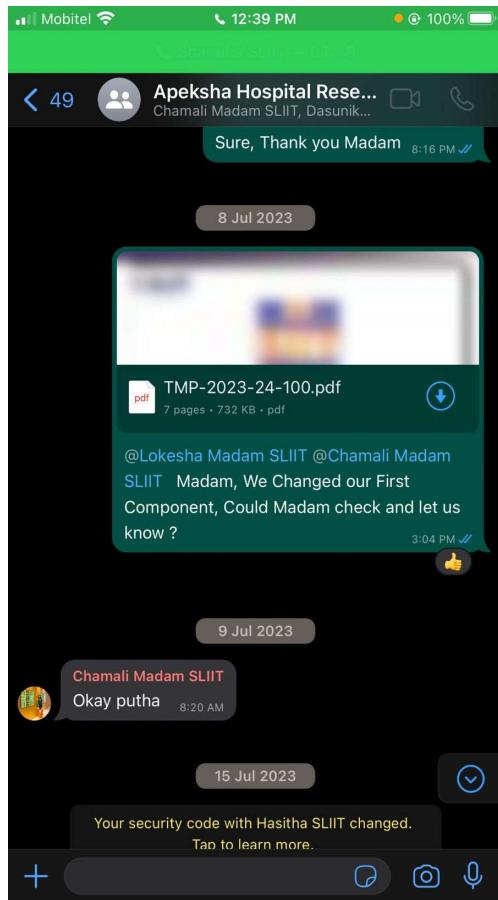
The image displays two screenshots of Microsoft Teams conversations, likely from a mobile device, showing research meeting logs.

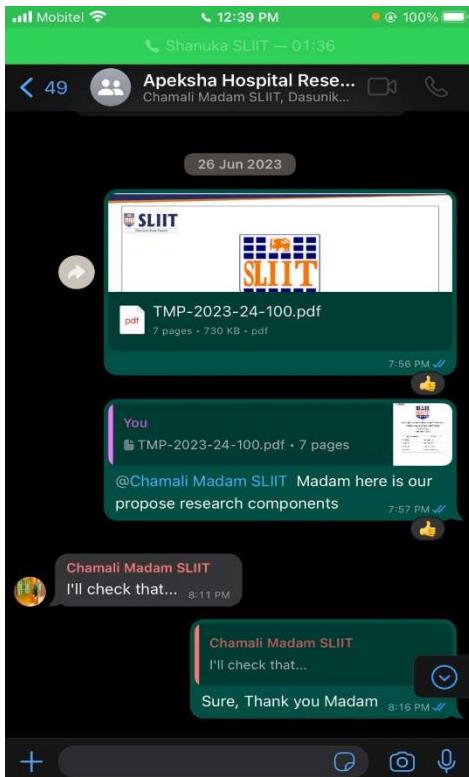
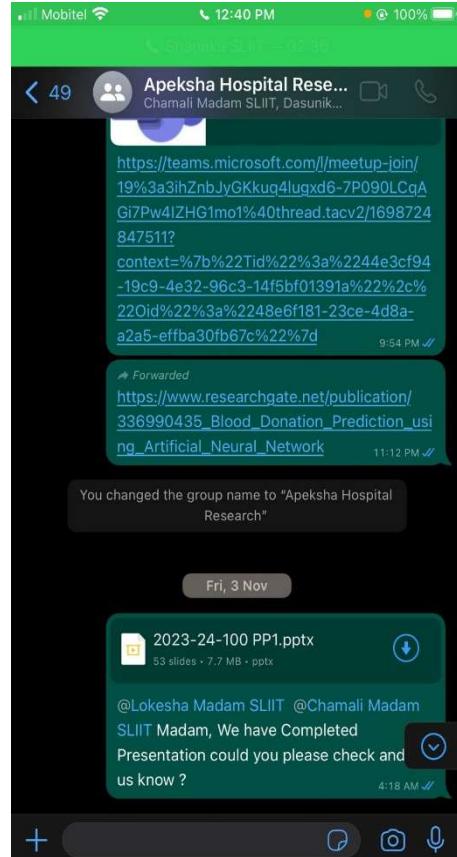
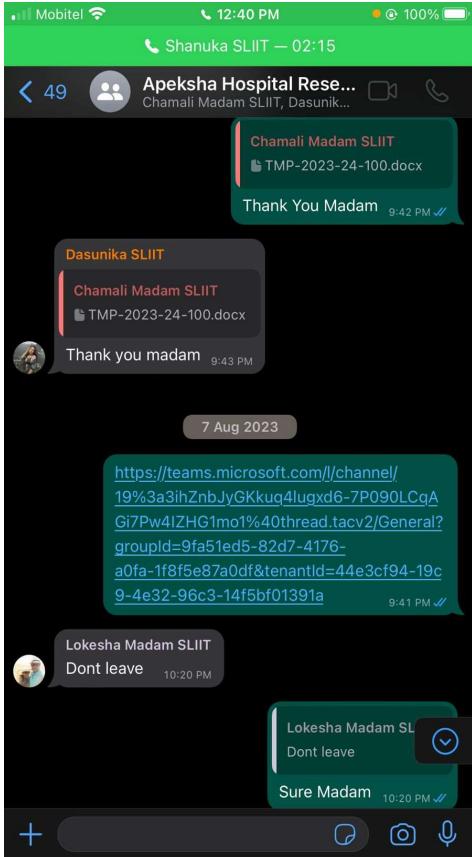
Screenshot 1 (Top): This screenshot shows a conversation in the "General" channel of the team "TMP-2023-24-100". The messages include:

- Punchihewa S.N it20665166 6/15/2023 12:21 PM: TMP-2023-24-100.pdf
- Punchihewa S.N it20665166 6/26/2023 7:55 PM: TMP-2023-24-100 (1).pdf
- Punchihewa S.N it20665166 6/26/2023 7:55 PM: Chamali Pathasari here is our propose research components
- Punchihewa S.N it20665166 8/7/2023 10:38 PM: Meeting in "General" ended
- Punchihewa S.N it20665166 8/23/2023 9:30 AM: Research Meeting Wednesday, August 23, 2023 9:30 PM

Screenshot 2 (Bottom): This screenshot shows a conversation in the "General" channel of the team "TMP-2023-24-100". The messages include:

- Punchihewa S.N it20665166 8/23/2023 10:03 PM: Scheduled a meeting
- Punchihewa S.N it20665166 8/23/2023 10:14 PM: 8/23/2023 10:14 PM - Meeting ended: 46m 15s
- Punchihewa S.N it20665166 10/31/2023 9:30 AM: Research 2023-24-100 Tuesday, October 31, 2023 9:30 PM
- Punchihewa S.N it20665166 10/31/2023 11:15 PM: Scheduled a meeting
- Punchihewa S.N it20665166 11/7/2023 8:00 PM: Apelsha Hospital Research - 2023-24-100 Wednesday, November 1, 2023 10:00 PM





Personal backlog

Try the New Boards Hub for improved performance, accessibility, and new features. [Click here](#) to learn more.

Apeksha Hospital Donor Engagement ... ★ ⓘ

Board Analytics View as Backlog

Filter by keyword Types Bandara H.R.H.S. it20662028 Tags Iteration Area Parent Work Item

New Active 3/5 Resolved 11/5 Closed

+ New item

32 Implement the Prediction Dashboard User Interface.
Bandara H.R.H.S. it20662028
State • New

18 Collecting Real Donation Campaign historical data from Apeksha Hospital
Bandara H.R.H.S. it20662028
State • Active

26 Creating test data set to feed the model
Bandara H.R.H.S. it20662028
State • Resolved

23 Pre-process the test Data set
Bandara H.R.H.S. it20662028
State • Resolved

31 Implementing the Attendees Prediction model using Random Forest Algorithm.
Bandara H.R.H.S. it20662028
State • Resolved

15 Collecting Test Data - Previous Donation campaign attendees count and date
Bandara H.R.H.S. it20662028
State • Closed

X

Overall Backlog

The screenshot shows the Azure DevOps Boards backlog for the 'Apeksha Hospital Donor Engagement System-2023-24-100 Team'. The backlog is organized into four columns: New, Active, Resolved, and Closed. The 'New' column contains items 44, 29, 30, and 32. The 'Active' column contains items 46, 48, 47, 14, 18, and 17. The 'Resolved' column contains items 53, 52, 51, 50, 19, and 49. The 'Closed' column contains items 45, 15, 16, 13, and 17. Each item card includes a summary, status (Active, Resolved, Closed), and assigned developer.

Category	Item ID	Description	Status	Assigned To
New	44	Combine all the models to implement the final model	New	Wijesooriya P.L.P.G.D.S
New	29	Implement the Medication Dashboard User Interface.	New	Shanuka Prabodha
New	30	Implement items prediction dashboard	New	Punchihewa S.N. r20665166
New	32	Implement the Prediction Dashboard User Interface.	New	Bandara H.R.H.S. r20662028
Active	46	Model Implemented to identify Dryness of Hair	Active	Wijesooriya P.L.P.G.D.S
Active	48	Implement the Web Application with User-friendly Interface	Active	Wijesooriya P.L.P.G.D.S
Active	47	Model Implemented to identify bleached of Hair	Active	Wijesooriya P.L.P.G.D.S
Active	14	Collecting Real Medication Data from Apeksha Hospital	Active	Shanuka Prabodha
Active	18	Collecting Real Donation Campign historical data from Apeksha Hospital	Active	Bandara H.R.H.S. r20662028
Active	17	Collecting Real Data from Apeksha Hospital for Intelligent Donor-Driven Inventory System for Essential Items	Active	
Resolved	53	Model Implemented to identify Dandruff & lice of Hair	Resolved	Wijesooriya P.L.P.G.D.S
Resolved	52	Model Implemented to identify length type of Hair	Resolved	Wijesooriya P.L.P.G.D.S
Resolved	51	Model Implemented to identify Color of Hair	Resolved	
Resolved	50	Data Pre-Processing and Create Data Set With Data Augmentation	Resolved	Wijesooriya P.L.P.G.D.S
Resolved	19	Creating test data set to feed the model	Resolved	Shanuka Prabodha
Resolved	49	Identification of best architecture for transfer learning	Resolved	Wijesooriya P.L.P.G.D.S
Closed	45	Collection Test Data- Promoting Quality Hair Donation for Cancer Patients	Closed	
Closed	15	Collecting Test Data - Previous Donation campaign attendees count and date	Closed	Bandara H.R.H.S. r20662028
Closed	16	Collection Test Data - Intelligent Donor-Driven Inventory System for Essential Items	Closed	Punchihewa S.N. r20665166
Closed	13	Collecting Test Data-Critical Medication Identification and shortage	Closed	Shanuka Prabodha
Closed	17	Collecting Real Data from Apeksha Hospital for Intelligent Donor-Driven Inventory System for Essential Items	Closed	

Trained Model Screenshot

The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** jupyter last Last Checkpoint: 11/02/2023 (autosaved)
- Toolbar:** File, Edit, View, Insert, Cell, Kernel, Widgets, Help
- Status Bar:** Not Trusted | Python 3 (ipykernel) | Logout
- Code Cell (In [1]):**

```
import pandas as pd
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import joblib

# Load the data
df = pd.read_csv('final_encoded_data.csv')

# Split the data into features (X) and the target (y)
x = df[['Month', "Day_Type_poyaday", "Day_Type_public", "Day_Type_weekday", "Day_Type_weekend']]
y = df['Number_of_Attendees']

# Create a random forest regressor
rf_regressor = RandomForestRegressor(n_estimators=100)

# Train the random forest regressor on the entire dataset
rf_regressor.fit(X, y)

# Save the trained model to a joblib file
joblib.dump(rf_regressor, 'random_forest_model.joblib')

# Get user input for prediction
user_month = int(input("Enter the month (1-12): "))
user_day_type_poyaday = int(input("Is it a Poyaday? (0 or 1): "))
user_day_type_public = int(input("Is it a public holiday? (0 or 1): "))
user_day_type_weekday = int(input("Is it a weekday? (0 or 1): "))
user_day_type_weekend = int(input("Is it a weekend? (0 or 1): "))

user_input = pd.DataFrame({
    "Month": [user_month],
    "Day_Type_poyaday": [user_day_type_poyaday],
    "Day_Type_public": [user_day_type_public],
    "Day_Type_weekday": [user_day_type_weekday],
    "Day_Type_weekend": [user_day_type_weekend]
}))
```
- Taskbar:** Shows various application icons like Spotify, Edge, and File Explorer.
- System Tray:** Displays system status icons (Wi-Fi, battery, etc.) and the date/time (2/23/2024, 2:59 PM).

Apeksha Hospital Donor Engagement System

2023-24-100

Status Document 1

Wijesooriya PLPGDS

BSc (Hons) in Information Technology specializing in Software Engineering

Department of Information Technology

Sri Lanka Institute of Information Technology Sri Lanka

August 2023

Promoting Quality Hair Donation for Cancer Patients

2023-24-100

Status Document 1

Wijesooriya PLPGDS

BSc (Hons) in Information Technology specializing in Software Engineering

Department of Information Technology

Sri Lanka Institute of Information Technology Sri Lanka

August 2023

Declaration of The Candidate & Supervisor

I declare that this is my work, and this proposal does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any other university or institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgment is made in the text.

Name	Student ID	Signature
Wijesooriya PLPGDS	IT20660352	

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

Signature of the supervisor:

Date:

Signature of the Co-supervisor:

Date:

Contents

1. Introduction	IV
2. Gantt Chart.....	V
3. Work-Break-Down Chart	VI
4. Screenshot of Meetings and Conversations.....	VI
5. Project Management Tools & Screenshot.....	XI
5.1 Overall Backlog.....	XI
5.2 Personal Backlog	XII
6. Trained Model Screenshot	XII
6.1 .Preprocessing for the image data sets	XII
6.2.Model for the Hair Type Detection	XIII
6.3.Model for the Hair Color Detection	XV
6.4. Model for the Dandruff & Lice Detection	XVII

1. Introduction

In the realm of cancer care, maintaining a sense of normalcy and confidence is paramount for patients undergoing treatment. One aspect often overlooked yet crucial for their well-being is the provision of high-quality wigs or hairpieces. However, the absence of a structured hair donation program at Apeksha Hospital poses a significant challenge in ensuring the availability of suitable hairpieces for cancer patients. To address this gap, it is imperative to establish a robust framework for assessing and promoting the quality of donated hair.

The research problem at hand revolves around determining whether donated hair meets the stringent standards and qualities mandated by Apeksha Hospital. This entails evaluating various characteristics of donated hair, including length, absence of split ends, and overall condition. Traditional methods of inspection may not suffice in ensuring the requisite quality standards. Therefore, leveraging cutting-edge image processing techniques emerges as a promising avenue to enhance the efficacy and accuracy of hair quality assessment.

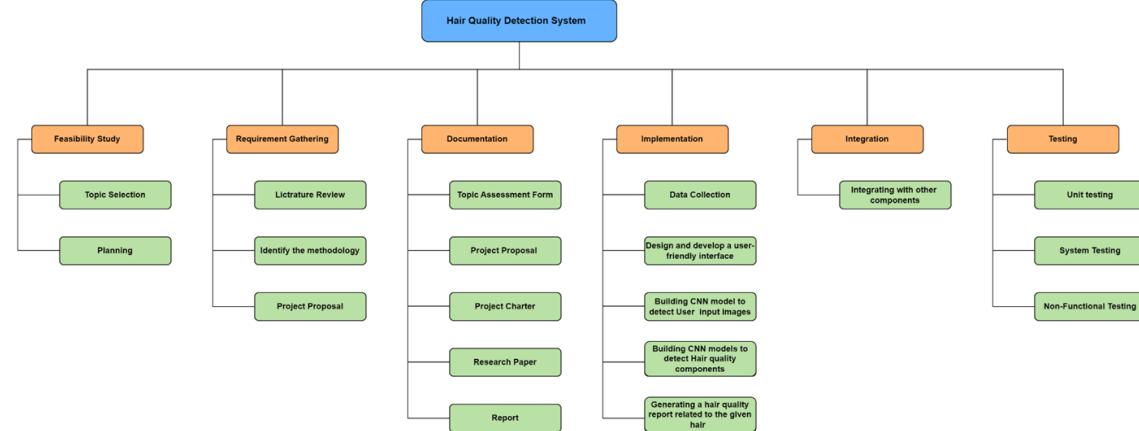
By employing image processing algorithms, donors can meticulously scrutinize donated hair for imperfections such as bleaching, dyeing, dryness, and split ends. This technology-driven approach not only streamlines the screening process but also minimizes the likelihood of substandard hair being utilized for crafting wigs or hairpieces. Moreover, promoting adherence to hospital standards fosters a culture of responsibility and accountability among potential donors, thereby augmenting the overall quality of donated hair.

In essence, this research endeavors to bridge the gap between hair donation and cancer patient care by establishing a comprehensive framework for evaluating and promoting the quality of donated hair. By integrating advanced image processing techniques with stringent quality criteria, Apeksha Hospital can ensure that cancer patients receive superior wigs or hairpieces, thereby enhancing their sense of confidence and well-being throughout the rigors of treatment.

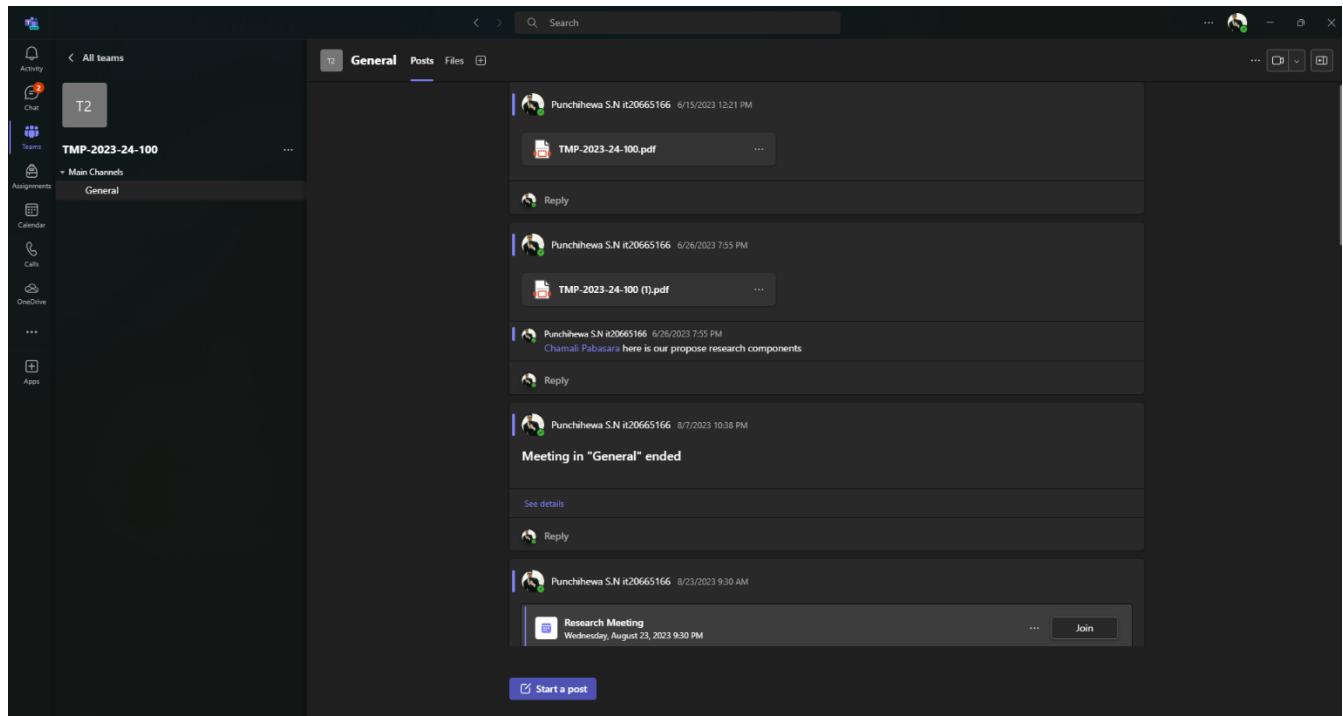
2. Gantt Chart

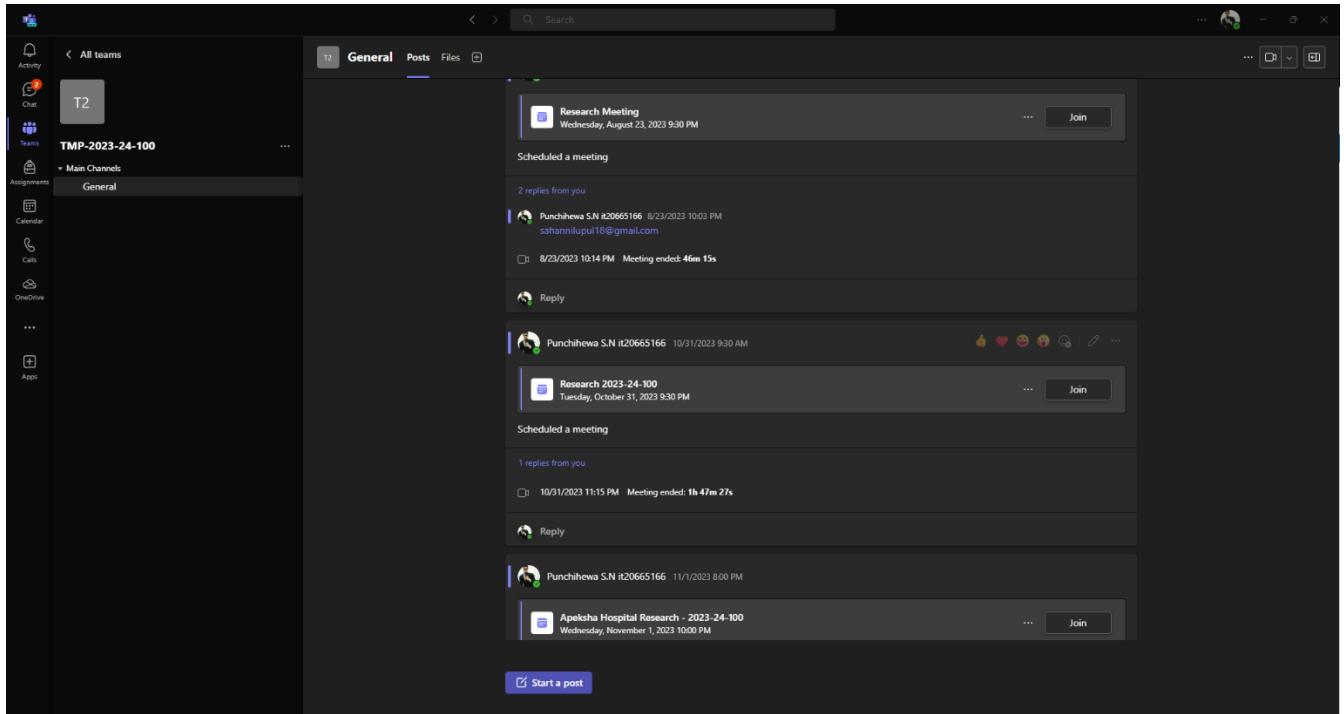
Task Name	Semester 1						Semester 2					
	June	July	August	September	October	November	December	January	February	March	April	May
Feasibility Study												
Topic Selection	■											
Topic Evaluation	■	■										
Background Study	■	■	■									
Background Study and Literature Survey												
Proposal Draft												
Project Proposal			■	■								
Proposal Presentation												
Proposal Report			■	■								
Project Initiation		■		■								
Data gathering			■	■								
Selecting technologies				■								
Project Implementation					■	■	■	■	■	■	■	
Data Collection					■	■	■	■	■	■	■	
Design and Develop a User friendly interface					■	■	■	■	■	■	■	
Building CNN Model to detect user input images					■	■	■	■	■	■	■	
Building CNN models to detect hair quality components					■	■	■	■	■	■	■	
generating a hair quality report related to the given hair					■	■	■	■	■	■	■	
Final Implementation					■	■	■	■	■	■	■	
Unit Testing						■	■	■	■	■	■	
Evaluating and error fixing						■	■	■	■	■	■	
Integrating with other components						■	■	■	■	■	■	
Final Stages							■	■	■	■	■	
System Integration							■	■	■	■	■	
System Testing							■	■	■	■	■	
Evaluating and error fixing							■	■	■	■	■	
Final Report								■	■	■	■	

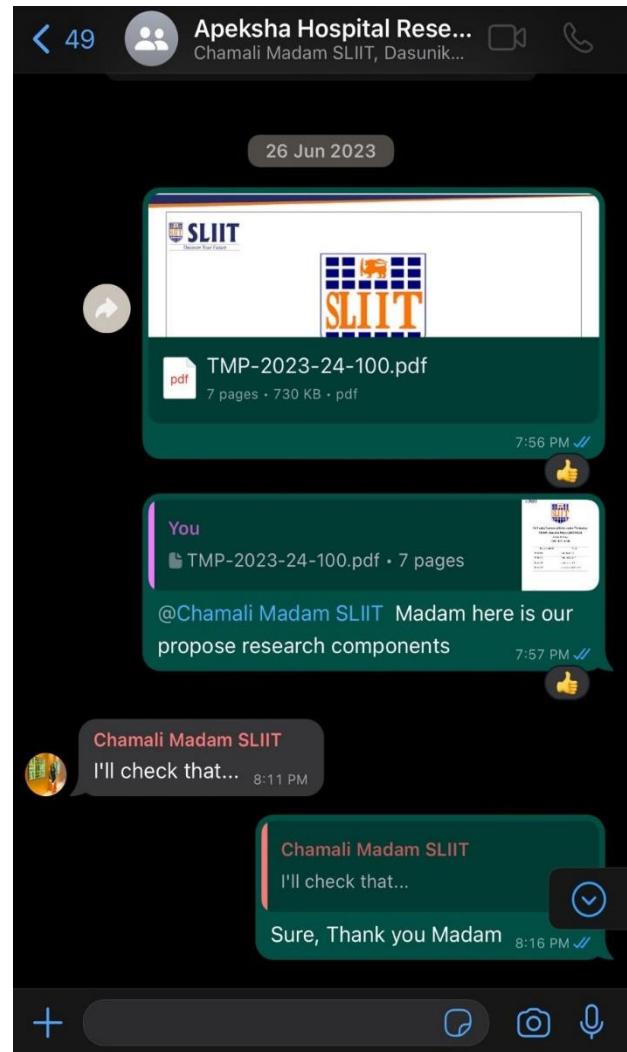
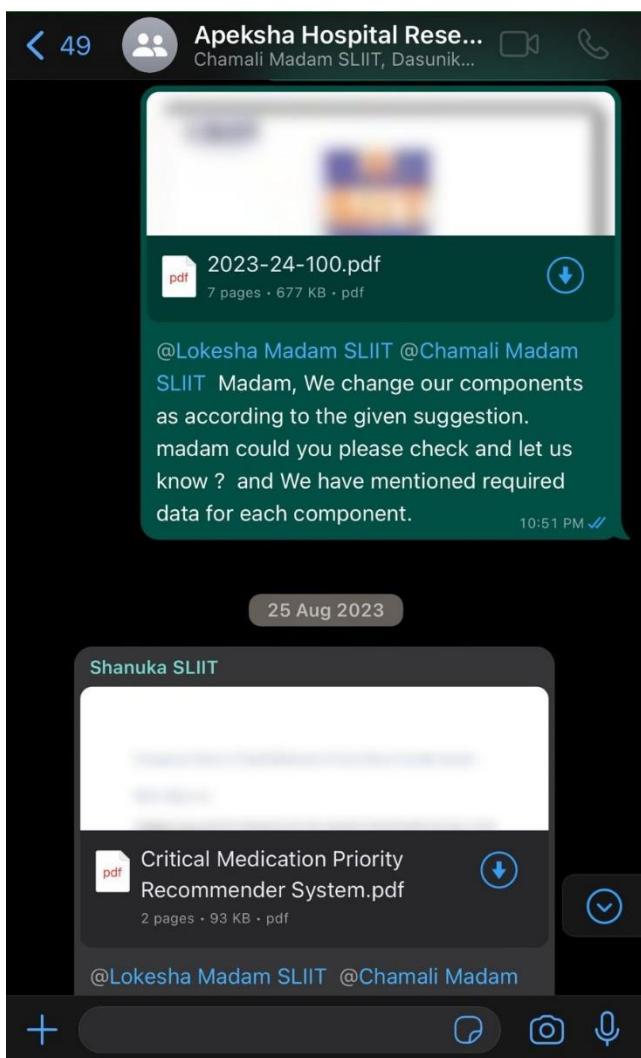
3. Work-Break-Down Chart

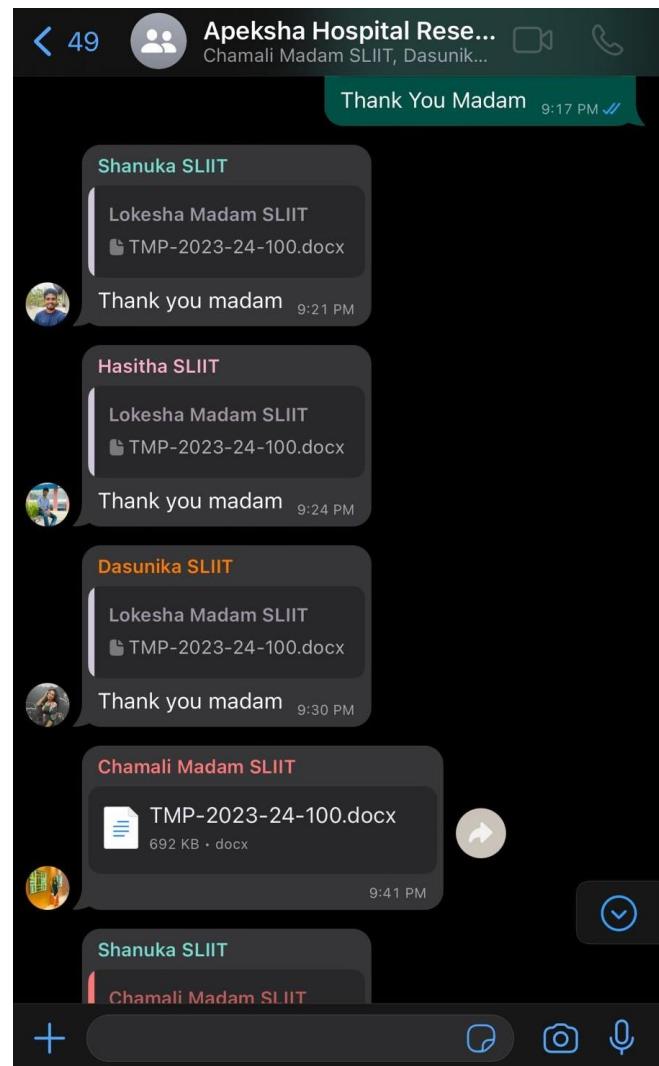
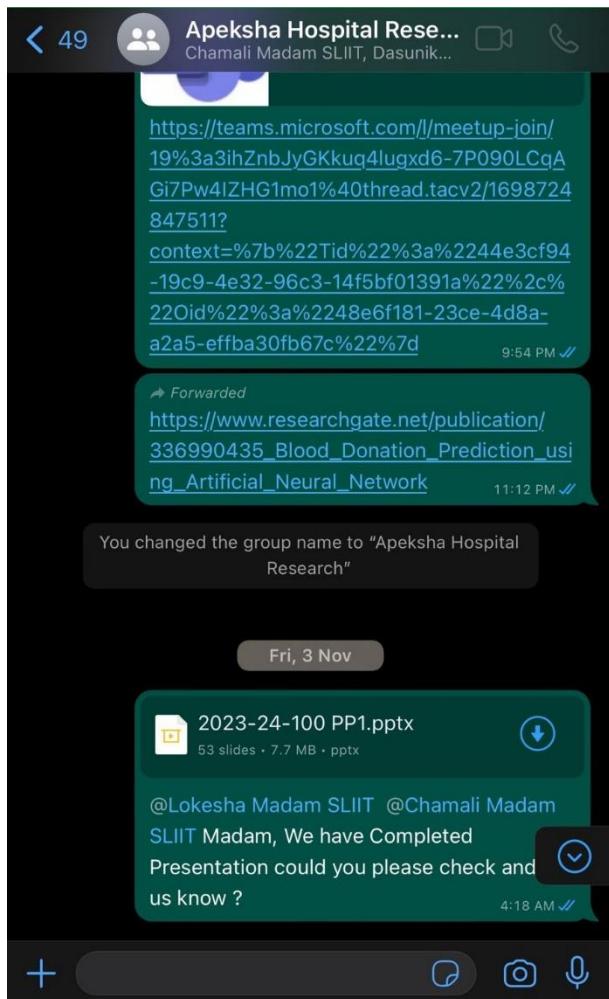


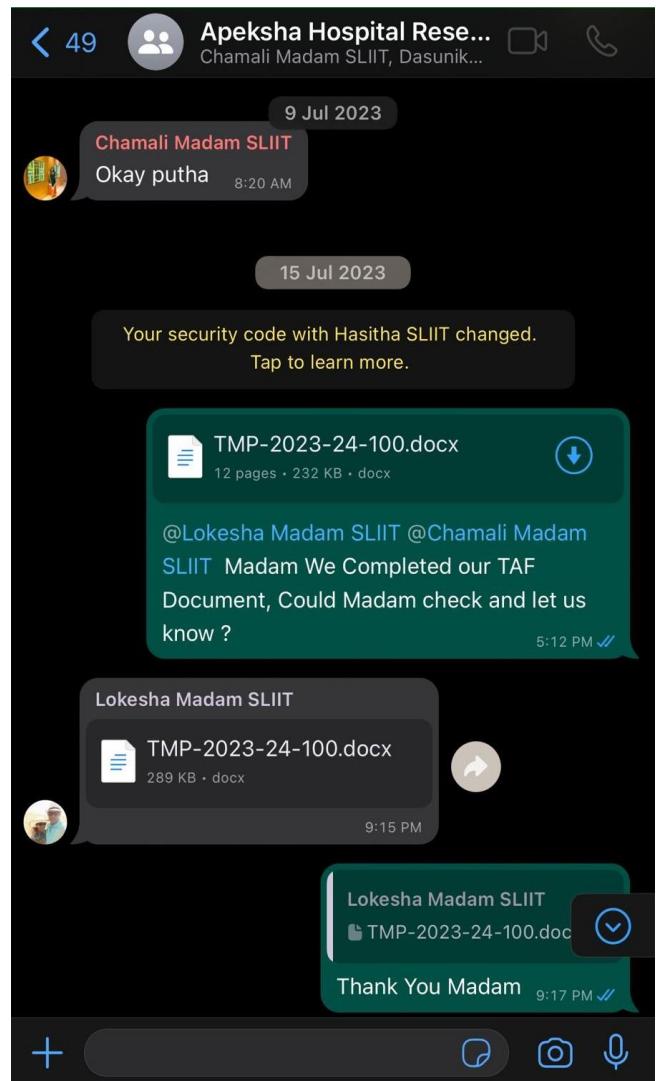
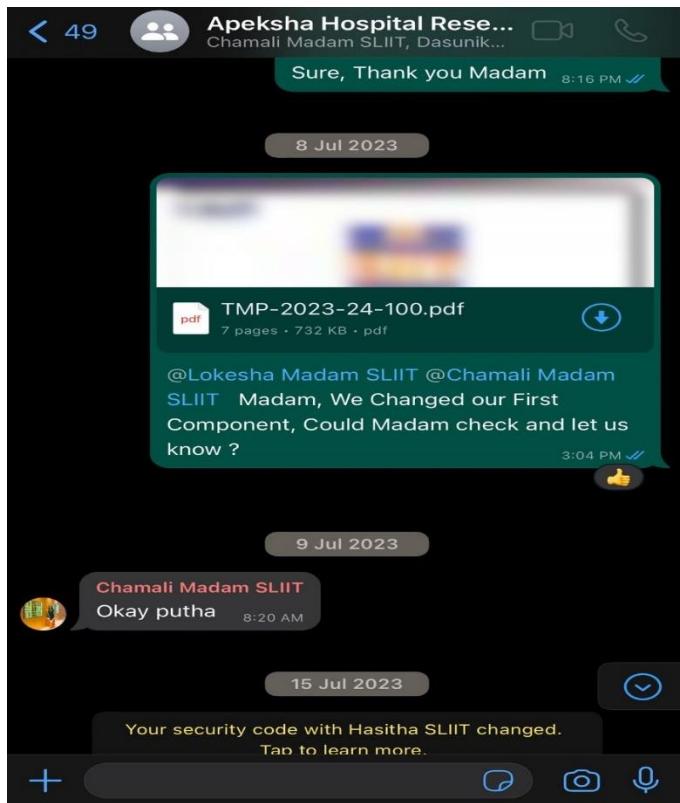
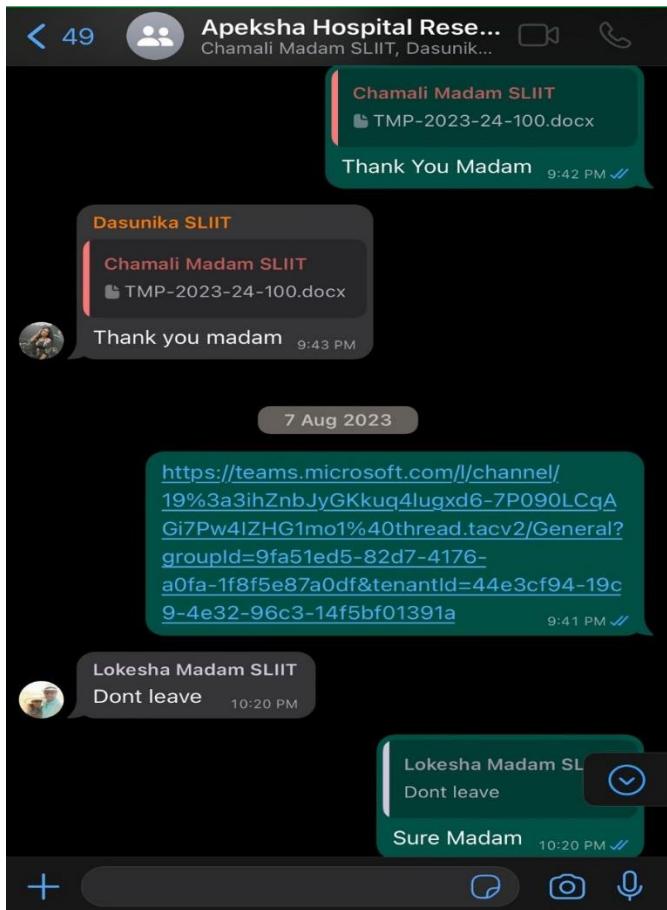
4. Screenshot of Meetings and Conversations











5. Project Management Tools & Screenshot

5.1 Overall Backlog

The screenshot shows the Azure DevOps Boards backlog for the "Apeksha Hospital Donor Engagement System-2023-24-100 Team". The backlog is organized into four columns: New, Active, Resolved, and Closed. Each column contains several items, each represented by a card with a summary, status (Active, Resolved, Closed), and assignee.

Column	Item Summary	Status	Assignee
New	44 Combine all the models to implement the final model	New	Wijesooriya P.L.P.G.D.S
New	29 Implement the Medication Dashboard User Interface	New	Shanuka Prabodha
New	30 Implement items prediction dashboard	New	Punichihewa S.N #20665166
New	32 Implement the Prediction Dashboard User Interface	New	Bandara H.R.H.S. #20662028
Active	46 Model Implemented to identify Dryness of Hair	Active	Wijesooriya P.L.P.G.D.S
Active	48 Implement the Web Application with User-friendly Interface	Active	Wijesooriya P.L.P.G.D.S
Active	47 Model Implemented to identify bleached of Hair	Active	Wijesooriya P.L.P.G.D.S
Active	14 Collecting Real Medication Data from Apeksha Hospital	Active	Shanuka Prabodha
Active	18 Collecting Real Donation Campaign historical data from Apeksha Hospital	Active	Bandara H.R.H.S. #20662028
Active	17 Collecting Real Data from Apeksha Hospital for Intelligent Donor-Driven Inventory System for Essential Items	Active	
Resolved	53 Model Implemented to identify Dandruff & lice of Hair	Resolved	Wijesooriya P.L.P.G.D.S
Resolved	52 Model Implemented to identify length type of Hair	Resolved	Wijesooriya P.L.P.G.D.S
Resolved	51 Model Implemented to identify Color of Hair	Resolved	
Resolved	50 Data Pre Processing and Create Data Set With Data Augmentation	Resolved	Wijesooriya P.L.P.G.D.S
Resolved	19 Creating test data set to feed the model	Resolved	Shanuka Prabodha
Resolved	49 Identification of best architecture for transfer learning	Resolved	Wijesooriya P.L.P.G.D.S
Closed	45 Collection Test Data - Promoting Quality Hair Donation for Cancer Patients	Closed	
Closed	15 Collecting Test Data - Previous Donation campaign attendees count and date	Closed	Bandara H.R.H.S. #20662028
Closed	16 Collection Test Data - Intelligent Donor-Driven Inventory System for Essential Items	Closed	Punichihewa S.N #20665166
Closed	13 Collecting Test Data-Critical Medication Identification and shortage	Closed	Shanuka Prabodha

5.2 Personal Backlog

The screenshot shows the Azure DevOps Boards backlog page for the project "Apeksha Hospital Donor Engagement". The backlog is organized into columns: New, Active, Resolved, and Closed. There are 3 items in the New column, 5 in the Resolved column, and 5 in the Closed column. The items are numbered 39 to 43. Each item has a summary, a link, and a state indicator (Active, Resolved, or Closed). The sidebar on the left shows navigation options like Overview, Boards, Work items, Backlogs, Sprints, Queries, Delivery Plans, Analytics views, Repos, Pipelines, and Project settings.

6. Trained Model Screenshot

6.1 Preprocessing for the image data sets

```
In [4]: import cv2
import img HDR

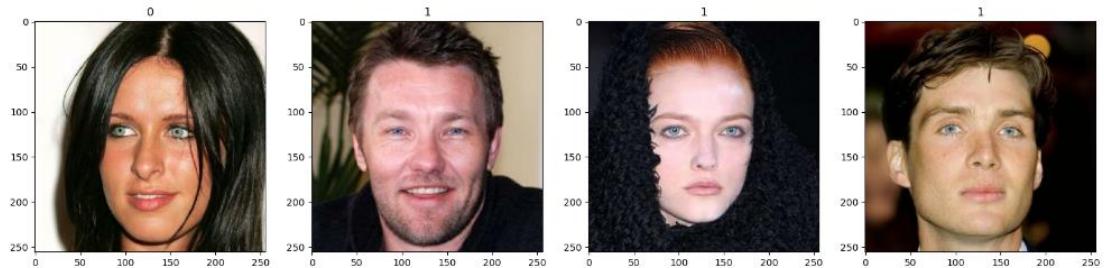
In [6]: image_exts = ['jpeg', 'jpg', 'bmp', 'png']

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 = img HDR.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 [ ]: data gen = preprocessing.image.ImageDataGenerator(
    rescale=1./255,
    validation_split=0.2
```

6.2 Model for the Hair Type Detection

```
In [16]: data_iterator = data.as_numpy_iterator()  
  
In [17]: batch = data_iterator.next()  
  
In [18]: 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])
```

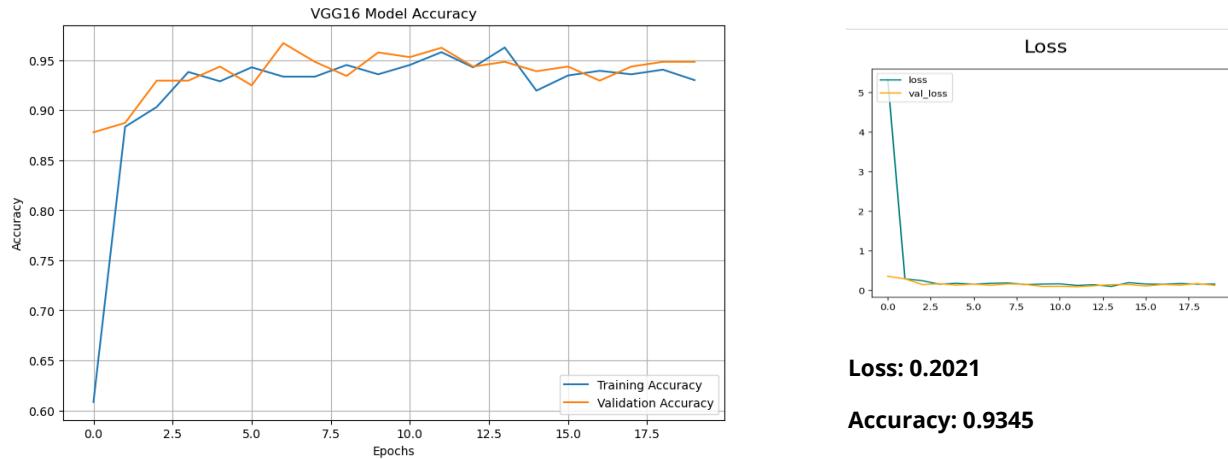


Proposed Model

6.2.1. CNN Architecture Model- VGG1

```
In [4]: train_generator = datagen.flow_from_directory(  
    'C:/Users/Ridma/Downloads/Hairs',  
    target_size=img_size,  
    batch_size=batch_size,  
    class_mode='categorical',  
    subset='training'  
)  
Found 858 images belonging to 2 classes.  
  
In [5]: val_generator = datagen.flow_from_directory(  
    'C:/Users/Ridma/Downloads/Hairs',  
    target_size=img_size,  
    batch_size=batch_size,  
    class_mode='categorical',  
    subset='validation'  
)  
Found 213 images belonging to 2 classes.  
  
In [6]: from tensorflow.keras.callbacks import EarlyStopping  
early_stop = EarlyStopping(monitor='val_loss', patience=10, verbose=1, restore_best_weights=True)  
  
In [7]: base_model = VGG16(input_shape=(160, 160, 3), include_top=False, weights='imagenet')  
base_model.trainable = False  
  
vgg16_model = models.Sequential([  
    base_model,  
    layers.Flatten(),  
    layers.Dense(4096, activation='relu'),  
    layers.Dropout(0.5),  
    layers.Dense(4096, activation='relu'),  
    layers.Dropout(0.5),  
    layers.Dense(2, activation='softmax')  
)  
  
vgg16_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])  
  
history=vgg16_model.fit(train_generator, validation_data=val_generator, epochs=20 , callbacks=[early_stop])  
Epoch 1/20  
27/27 [=====] - 163s 6s/step - loss: 5.3365 - accuracy: 0.6084 - val_loss: 0.3546 - val_accuracy: 0.87
```

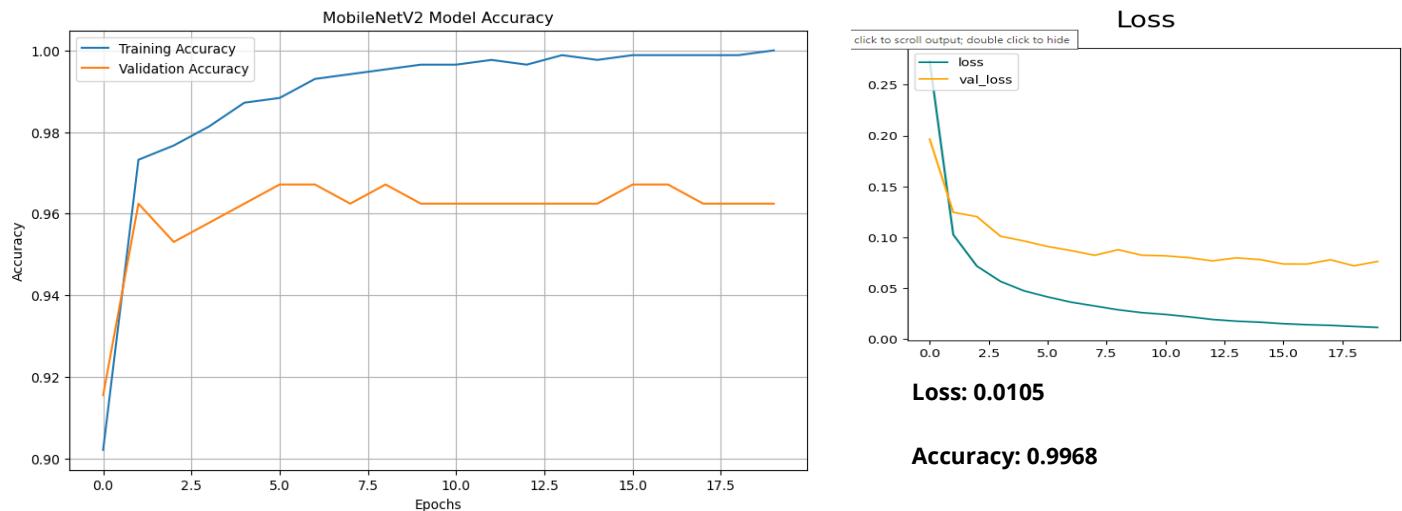
Evidence of Completion



6.2.2. CNN Architecture Model- MobileNetV2

```
In [38]: train_generator = datagen.flow_from_directory(  
    'C:/Users/Ridma/Downloads/Hairs',  
    target_size=img_size,  
    batch_size=batch_size,  
    class_mode='categorical',  
    subset='training'  
)  
Found 858 images belonging to 2 classes.  
  
In [39]: val_generator = datagen.flow_from_directory(  
    'C:/Users/Ridma/Downloads/Hairs',  
    target_size=img_size,  
    batch_size=batch_size,  
    class_mode='categorical',  
    subset='validation'  
)  
Found 213 images belonging to 2 classes.  
  
In [40]: from tensorflow.keras.callbacks import EarlyStopping  
early_stop = EarlyStopping(monitor='val_loss', patience=10, verbose=1, restore_best_weights=True)  
  
In [10]: base_model = MobileNetV2(input_shape=(160, 160, 3), include_top=False, weights='imagenet')  
base_model.trainable = False  
  
mobilenet_model = models.Sequential([  
    base_model,  
    layers.GlobalAveragePooling2D(),  
    layers.Dense(2, activation='softmax')  
)  
  
mobilenet_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])  
history=mobilenet_model.fit(train_generator, validation_data=val_generator, epochs=20 ,callbacks=[early_stop])  
  
Epoch 1/20  
14/14 [=====] - 26s 1s/step - loss: 0.2733 - accuracy: 0.9021 - val_loss: 0.1966 - val_accuracy: 0.915  
5  
Epoch 2/20  
14/14 [=====] - 16s 1s/step - loss: 0.1027 - accuracy: 0.9732 - val_loss: 0.1247 - val_accuracy: 0.962  
4
```

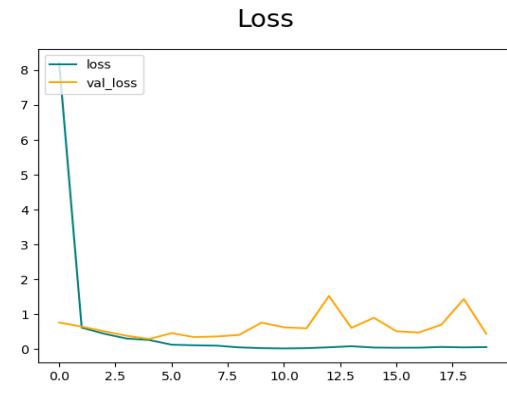
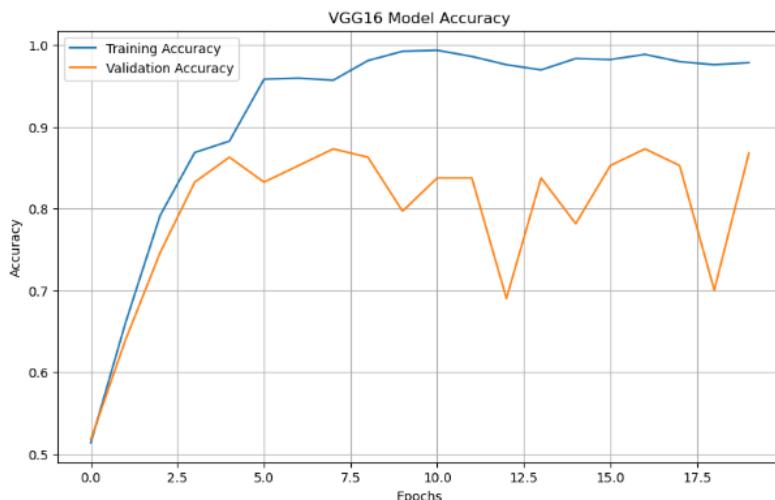
Evidence of Completion



6.3 Model for the Hair Color Detection

6.3.1. CNN Architecture Model- VGG16

```
In [6]: train_generator = datagen.flow_from_directory(  
    'C:/Users/Ridma/Downloads/newimages',  
    target_size=img_size,  
    batch_size=batch_size,  
    class_mode='categorical',  
    subset='training'  
)  
Found 792 images belonging to 2 classes.  
  
In [7]: val_generator = datagen.flow_from_directory(  
    'C:/Users/Ridma/Downloads/newimages',  
    target_size=img_size,  
    batch_size=batch_size,  
    class_mode='categorical',  
    subset='validation'  
)  
Found 197 images belonging to 2 classes.  
  
In [8]: base_model = VGG16(input_shape=(160, 160, 3), include_top=False, weights='imagenet')  
base_model.trainable = False  
  
vgg16_model = models.Sequential([  
    base_model,  
    layers.Flatten(),  
    layers.Dense(4096, activation='relu'),  
    layers.Dropout(0.5),  
    layers.Dense(4096, activation='relu'),  
    layers.Dropout(0.5),  
    layers.Dense(2, activation='softmax') # Assuming 2 hair colors  
])  
  
vgg16_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])  
history=vgg16_model.fit(train_generator, validation_data=val_generator, epochs=20)  
Epoch 1/20  
13/13 [=====] - 209s 16s/step - loss: 8.1964 - accuracy: 0.5139 - val_loss: 0.7592 - val_accuracy: 0.5
```



6.3.2. CNN Architecture Model- MobileNetV2

```
In [13]: train_generator = datagen.flow_from_directory(
    'C:/Users/Ridma/Downloads/newimages',
    target_size=img_size,
    batch_size=batch_size,
    class_mode='categorical',
    subset='training'
)
```

Found 792 images belonging to 2 classes.

```
In [14]: val_generator = datagen.flow_from_directory(
    'C:/Users/Ridma/Downloads/newimages',
    target_size=img_size,
    batch_size=batch_size,
    class_mode='categorical',
    subset='validation'
)
```

Found 197 images belonging to 2 classes.

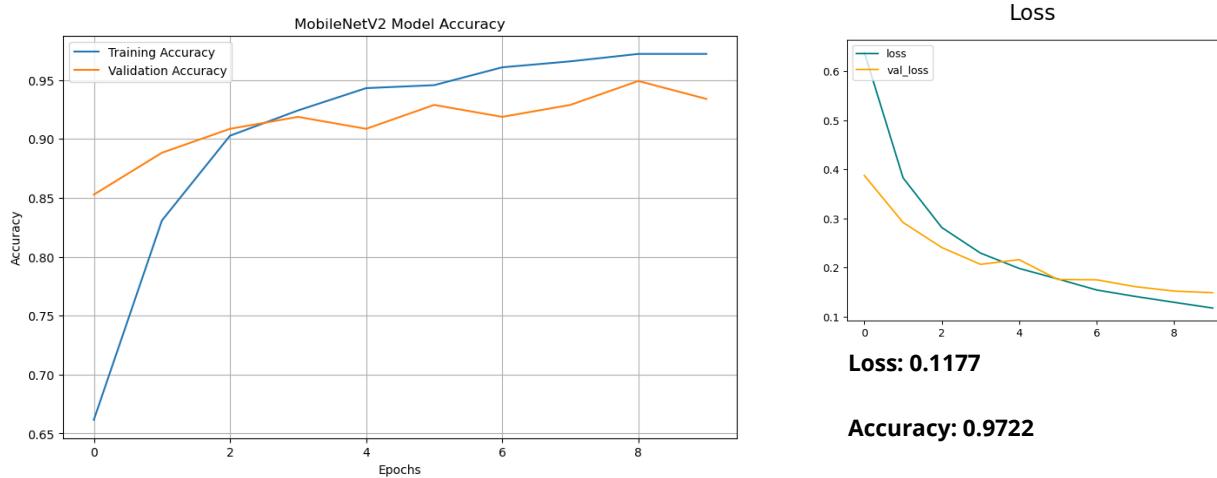
```
In [15]: base_model = MobileNetV2(input_shape=(160, 160, 3), include_top=False, weights='imagenet')
base_model.trainable = False

mobilenet_model = models.Sequential([
    base_model,
    layers.GlobalAveragePooling2D(),
    layers.Dense(2, activation='softmax') # Assuming 2 hair colors
])

mobilenet_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
history=mobilenet_model.fit(train_generator, validation_data=val_generator, epochs=10)
```

```
Epoch 1/10
13/13 [=====] - 42s 3s/step - loss: 0.6384 - accuracy: 0.6616 - val_loss: 0.3875 - val_accuracy: 0.852
8
```

Evidence of Completion

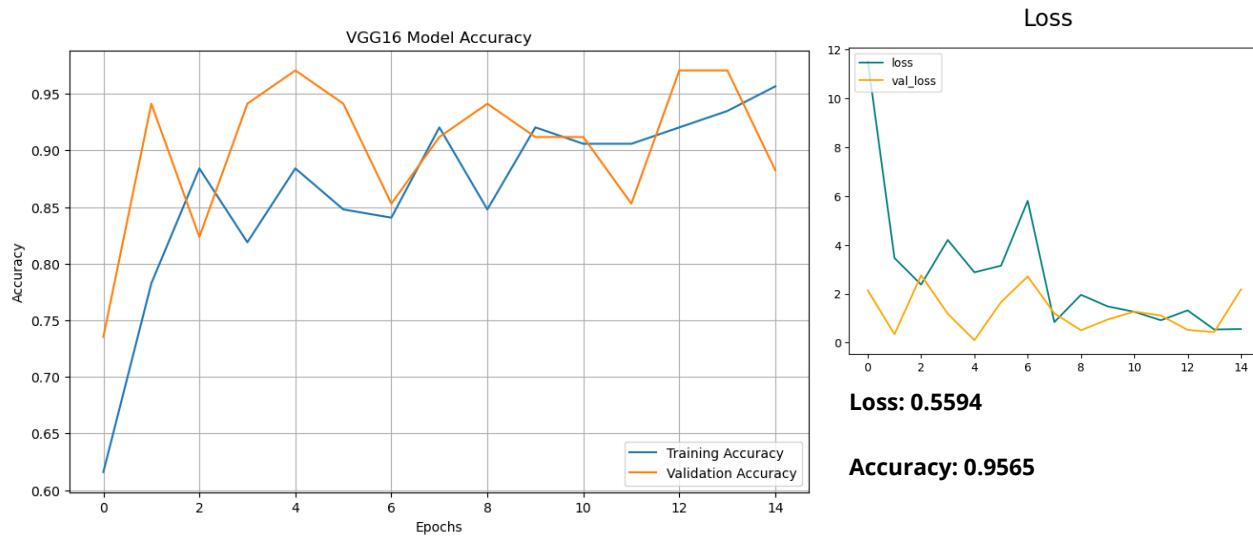


6.4. Model for the Dandruff & Lice Detection

6.4.1. CNN Architecture Model- VGG16

```
In [5]: train_generator = datagen.flow_from_directory(  
    'C:/Users/Ridma/Downloads/Dandruff',  
    target_size=img_size,  
    batch_size=batch_size,  
    class_mode='categorical',  
    subset='training'  
)  
Found 138 images belonging to 2 classes.  
  
In [6]: val_generator = datagen.flow_from_directory(  
    'C:/Users/Ridma/Downloads/Dandruff',  
    target_size=img_size,  
    batch_size=batch_size,  
    class_mode='categorical',  
    subset="validation"  
)  
Found 34 images belonging to 2 classes.  
  
In [7]: from tensorflow.keras.callbacks import EarlyStopping  
early_stop = EarlyStopping(monitor='val_loss', patience=10, verbose=1, restore_best_weights=True)  
  
In [8]: base_model = VGG16(input_shape=(160, 160, 3), include_top=False, weights='imagenet')  
base_model.trainable = False  
  
vgg16_model = models.Sequential([  
    base_model,  
    layers.Flatten(),  
    layers.Dense(4096, activation='relu'),  
    layers.Dropout(0.5),  
    layers.Dense(4096, activation='relu'),  
    layers.Dropout(0.5),  
    layers.Dense(2, activation='softmax') |  
)  
  
vgg16_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])  
history=vgg16_model.fit(train_generator, validation_data=val_generator, epochs=20 , callbacks=[early_stop])  
Epoch 1/20
```

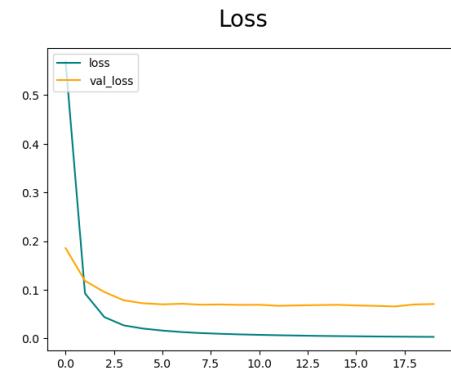
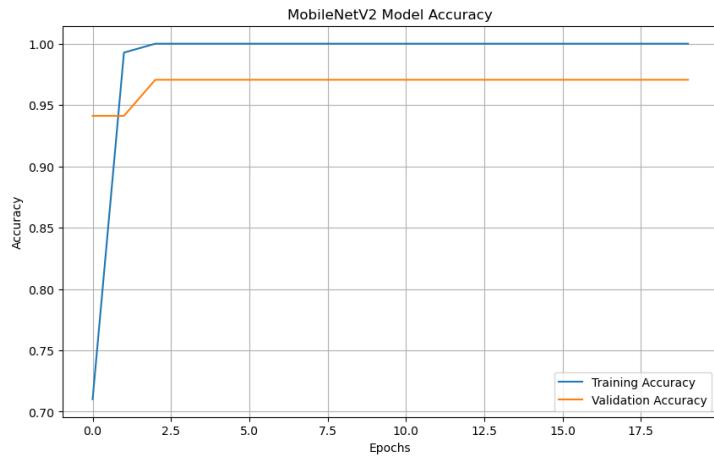
Evidence of Completion



6.4.2. CNN Architecture Model- MobileNetV2

```
In [25]: train_generator = datagen.flow_from_directory(  
    'C:/Users/Ridma/Downloads/Dandruff',  
    target_size=img_size,  
    batch_size=batch_size,  
    class_mode='categorical',  
    subset='training'  
)  
Found 138 images belonging to 2 classes.  
  
In [26]: val_generator = datagen.flow_from_directory(  
    'C:/Users/Ridma/Downloads/Dandruff',  
    target_size=img_size,  
    batch_size=batch_size,  
    class_mode='categorical',  
    subset='validation'  
)  
Found 34 images belonging to 2 classes.  
  
In [27]: from tensorflow.keras.callbacks import EarlyStopping  
early_stop = EarlyStopping(monitor='val_loss', patience=10, verbose=1, restore_best_weights=True)  
  
In [28]: base_model = MobileNetV2(input_shape=(160, 160, 3), include_top=False, weights='imagenet')  
base_model.trainable = False  
  
mobilenet_model = models.Sequential([  
    base_model,  
    layers.GlobalAveragePooling2D(),  
    layers.Dense(2, activation='softmax')  
)  
  
mobilenet_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])  
history=mobilenet_model.fit(train_generator, validation_data=val_generator, epochs=20, callbacks=[early_stop])  
Epoch 1/20  
18/18 [=====] - 18s 657ms/step - loss: 0.5687 - accuracy: 0.7101 - val_loss: 0.1855 - val_accuracy: 0.  
9412  
Epoch 2/20
```

Evidence of Completion



Loss:0.0032

Accuracy: 1.0000

