FLAREPATH – ADVANCED VEHICLE FIRE SAFETY AND MONITORING WITH RAPID EMERGENCY DISPATCH SOLUTIONS

R24-058

Status Document-1

Peramunage A N – IT21080562

B.Sc. (Hons) Degree in Information Technology specializing in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

March 2023

Group Details

Supervisor – Mr.Nelum Chathuranga Amarasena

Co-supervisor – Mr. Deemantha Nayanajith Siriwardana

External Supervisor – Mr. Onray Sahinda

Student Name	**Student ID**	**Contact No**	**Email Address**
Dharmagunawardana W.M.P.I	IT21132346	0772785361	it21132346@my.sliit.lk
Anthick G.N	IT21096266	0779820516	it21096266@my.sliit.lk
Abeywardhana D.N	IT21133718	0714057155	it21133718@my.sliit.lk
Peramunage A.N	IT21080562	0713999266	it21080562@my.sliit.lk

Table of Contents

Group Details **2**

Meetings & Calls **4**

Meetings with supervisor and co-supervisor 4

Meetings with Domain Experts 8

Snapshots from Field Visit **9**

Clickup Tasks Allocation **11**

CkickUp Dashboard 11

In Progress Tasks 12

Completed Tasks upto PP1 12

Project Implementation **13**

Data Collection 13

Machine Learning Model Training ..... Error! Bookmark not defined.

CNN Model ..... **Error! Bookmark not defined.**

RNN Model & SVM Model 18

Wireframes **24**

Mobile Application UI **27**

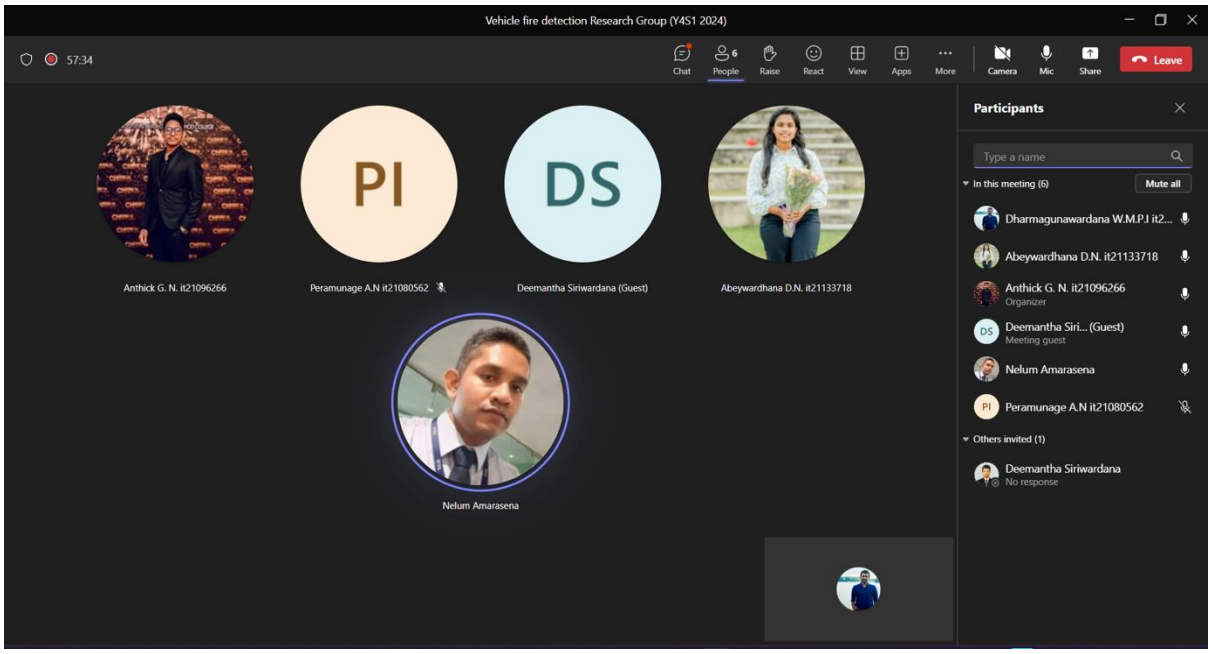
Gantt Chart **30**

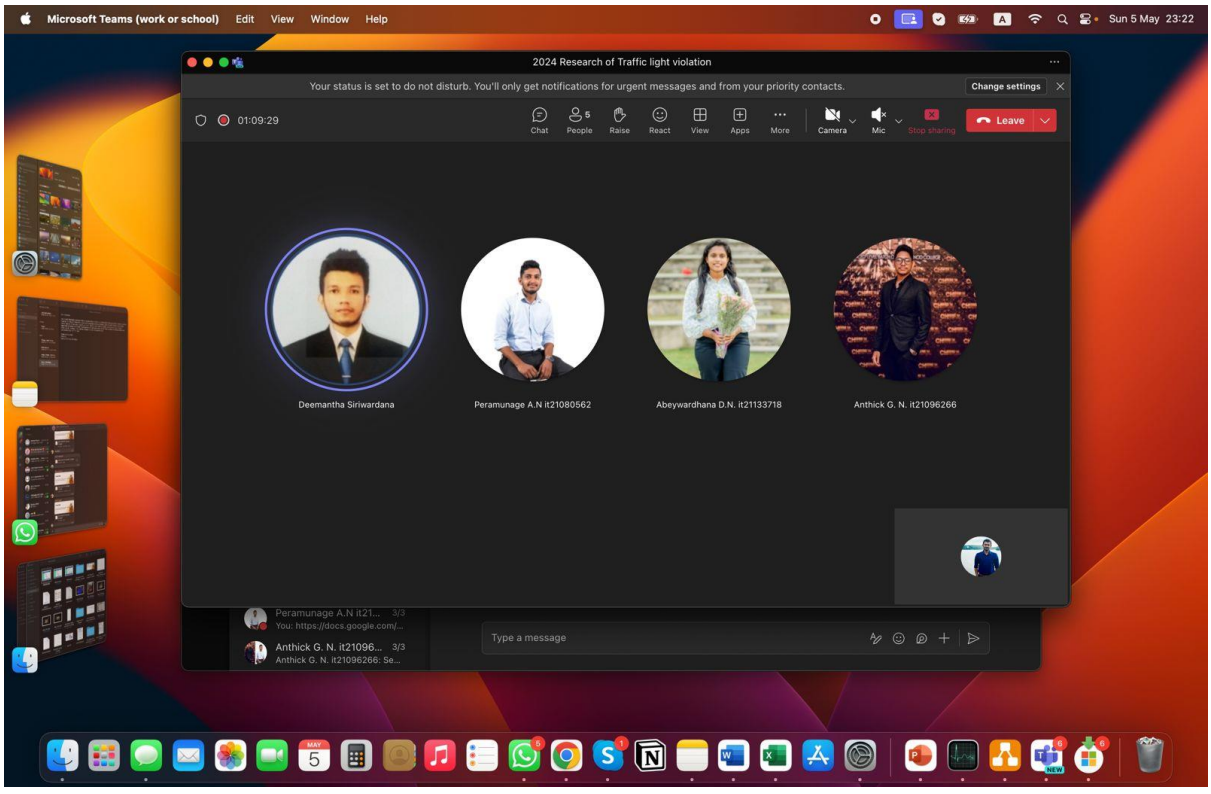
Work Breakdown Structure **30**

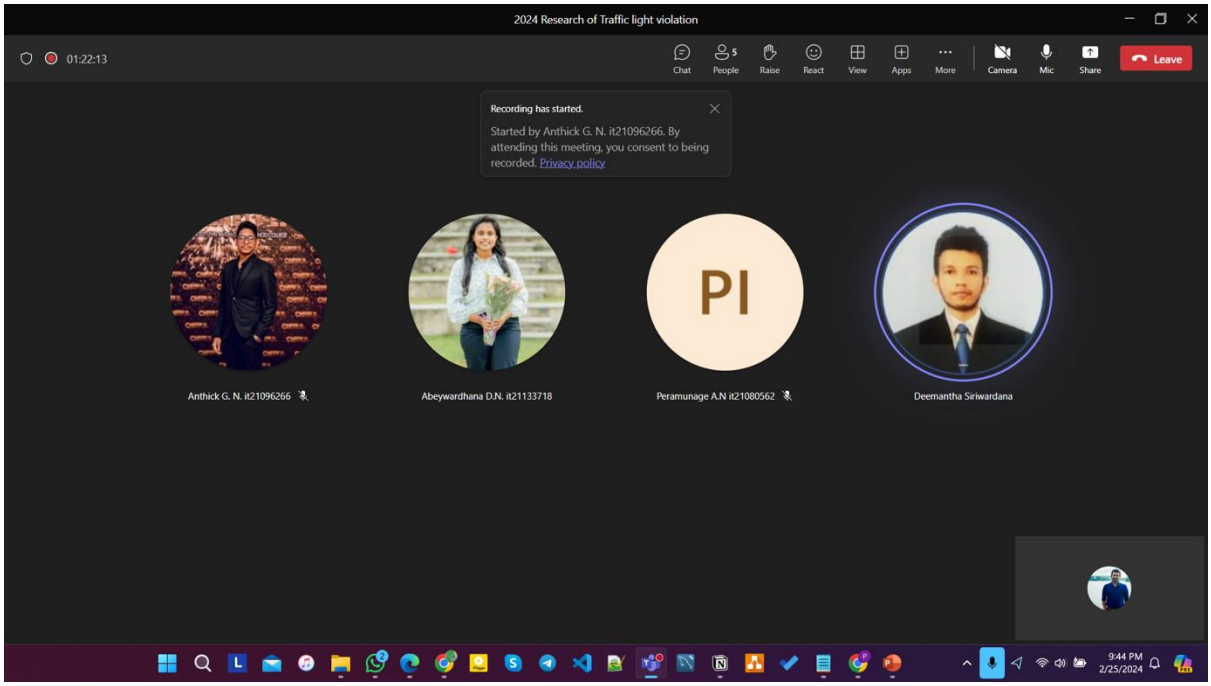
Meetings & Calls

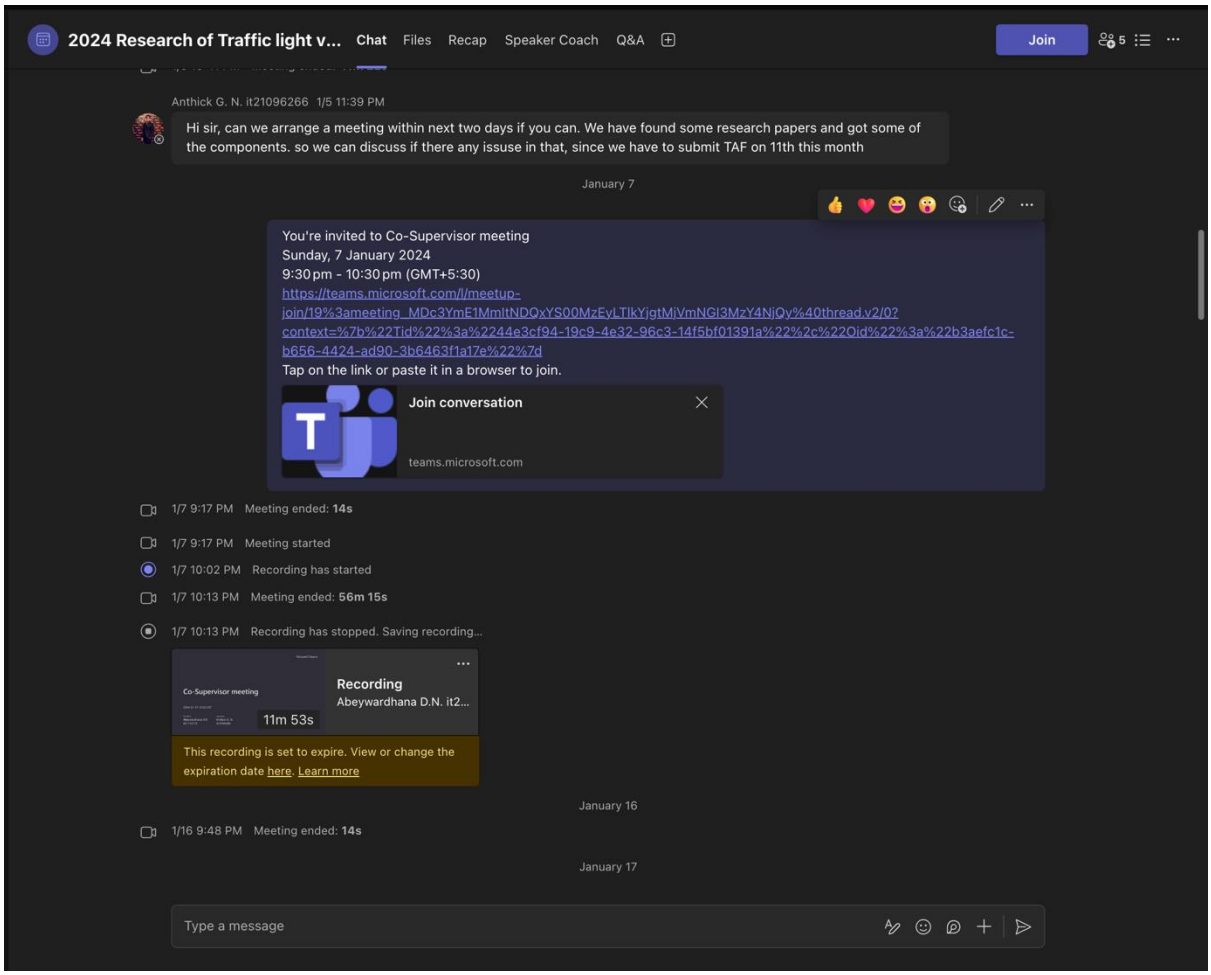
Meetings with supervisor and co-supervisor

Meeting with both supervisor and co-supervisor about the project progress and improvements that we need to do to our project.

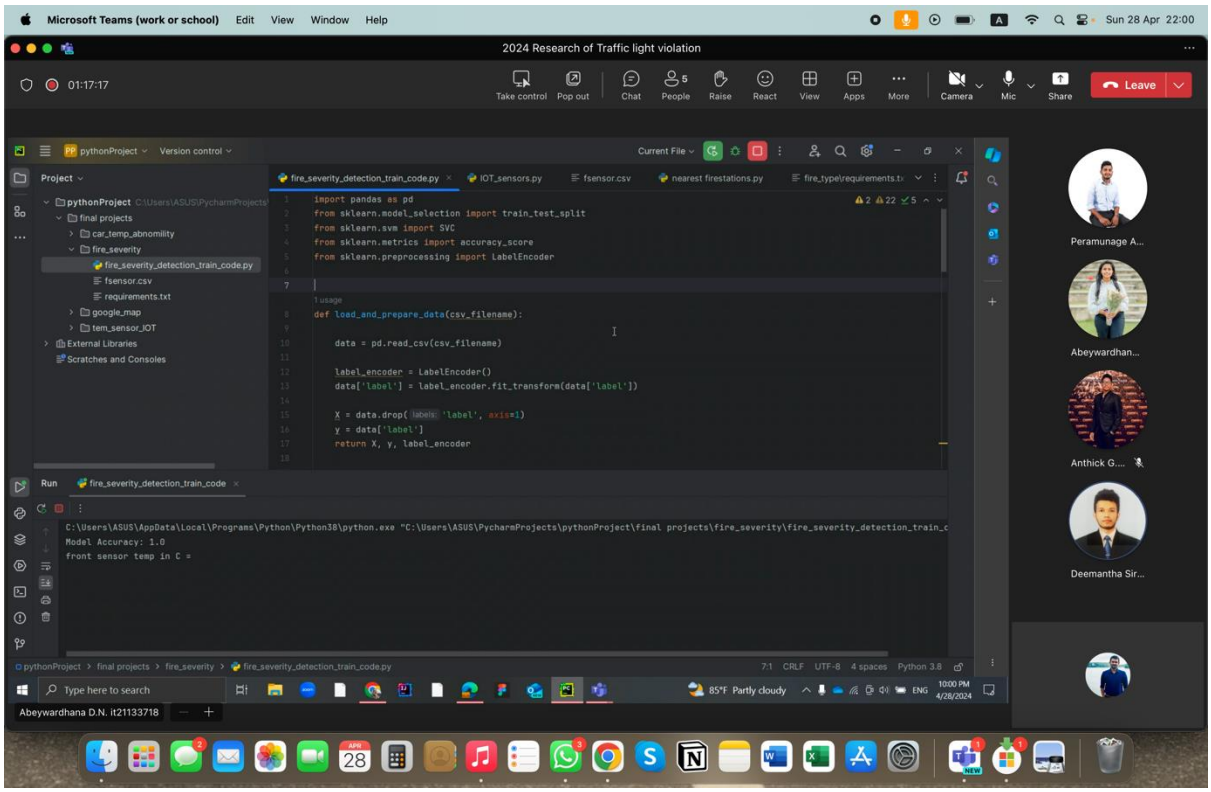




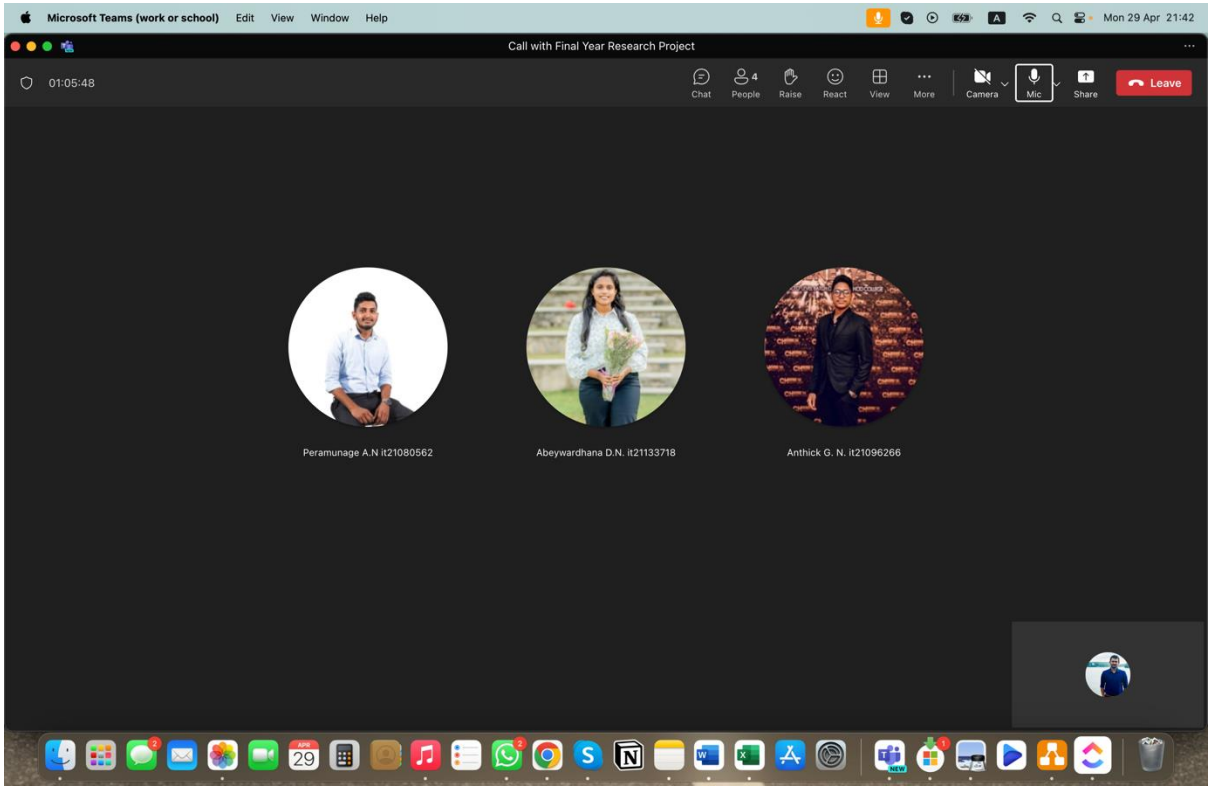

2024 Research of Traffic light violation
01:22:13
Chat
People
Raise
React
View
Apps
More
Camera
Mic
Share
Leave
Recording has started.
Started by Anthick G. N. it21096266. By attending this meeting, you consent to being recorded. Privacy policy
PI
Anthick G. N. it21096266
Abeywardhana D.N. it21133718
Peramunage A.N it21080562
Deemantha Siriwardana
9:44 PM
2/25/2024


2024 Research of Traffic light v...
Chat
Files
Recap
Speaker Coach
Q&A
Join
Anthick G. N. it21096266 1/5 11:39 PM
Hi sir, can we arrange a meeting within next two days if you can. We have found some research papers and got some of the components. so we can discuss if there any issuse in that, since we have to submit TAF on 11th this month
January 7
You're invited to Co-Supervisor meeting
Sunday, 7 January 2024
9:30 pm - 10:30 pm (GMT+5:30)
https://teams.microsoft.com/l/meetup-join/19%3ameeting_MDc3YmE1MmItNDQxYS00MzEyLTlkYjgtMjVmNGI3MzY4NjQy%40thread.v2/0?context=%7b%22Tid%22%3a%2244e3cf94-19c9-4e32-96c3-14f5bf01391a%22%2c%22Oid%22%3a%22b3aefc1c-b656-4424-ad90-3b6463f1a17e%22%7d
Tap on the link or paste it in a browser to join.
Join conversation
teams.microsoft.com
1/7 9:17 PM Meeting ended: 14s
1/7 9:17 PM Meeting started
1/7 10:02 PM Recording has started
1/7 10:13 PM Meeting ended: 56m 15s
1/7 10:13 PM Recording has stopped. Saving recording...
Co-Supervisor meeting
11m 53s
Recording
Abeywardhana D.N. it2...
This recording is set to expire. View or change the expiration date here. Learn more
January 16
1/16 9:48 PM Meeting ended: 14s
January 17
Type a message

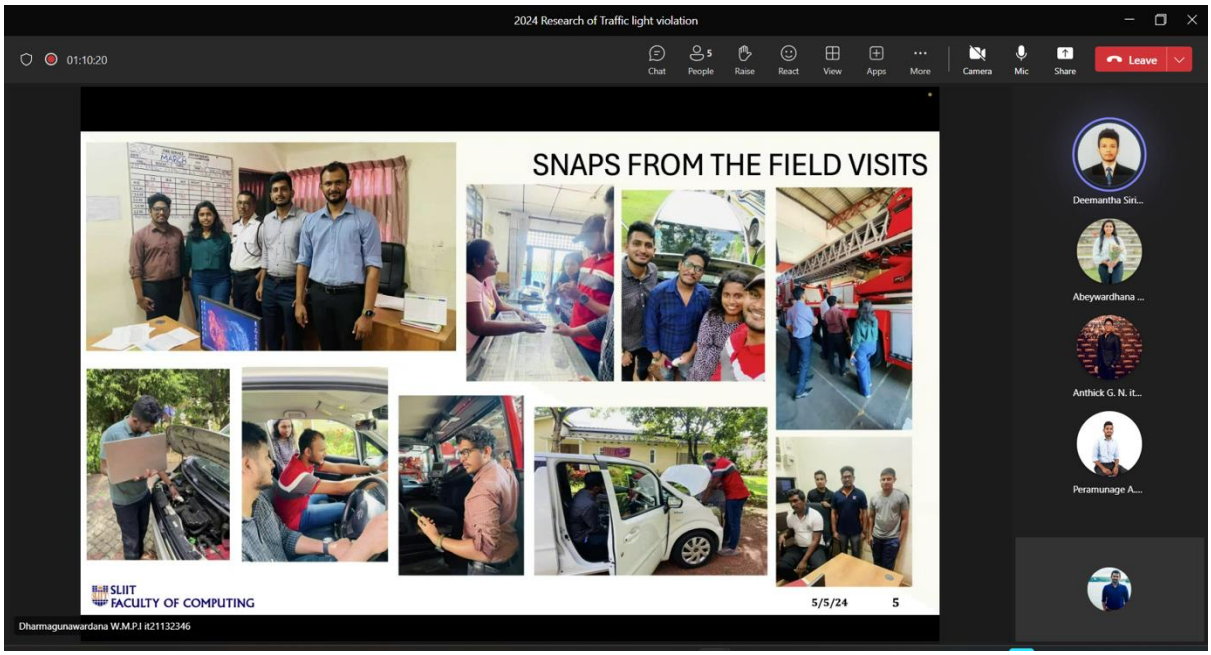
Code review with co-supervisor



Group meeting with group members



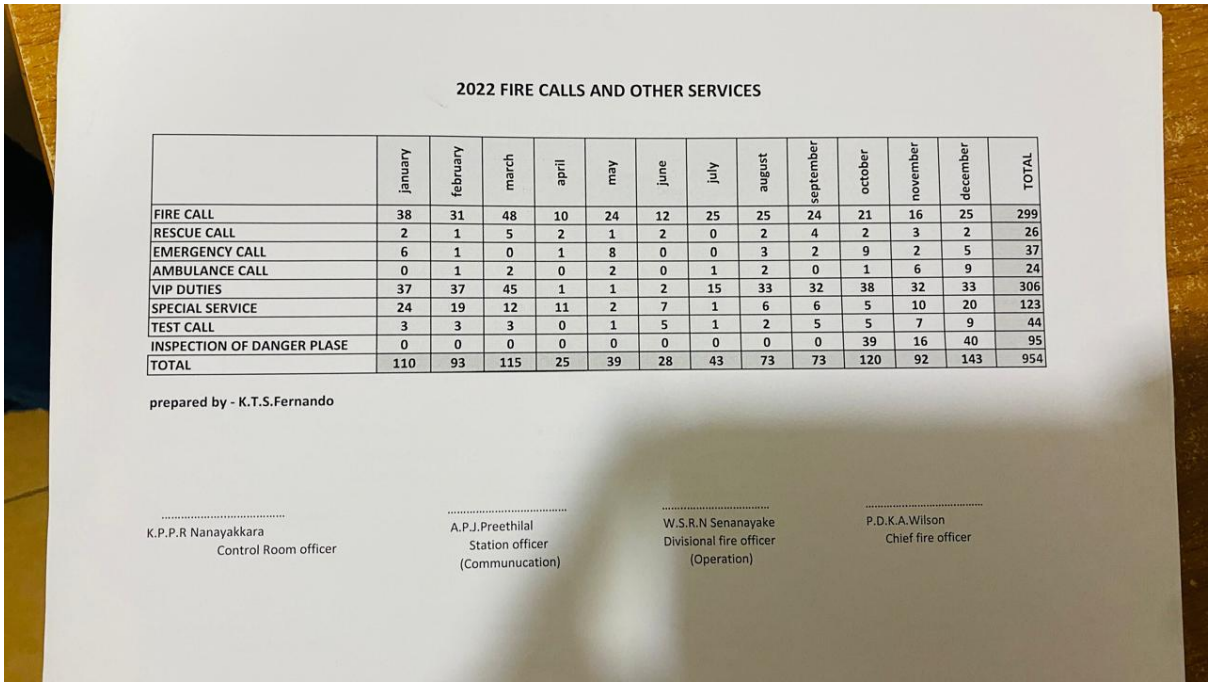
Presentation review with supervisor



Meetings with Domain Experts

Meeting With fire department officers and staff.

Mr.Nanayakkara the chief office of Fire department and we discussed about the domain knowledge and requirements.



2022 FIRE CALLS AND OTHER SERVICES

	january	february	march	april	may	june	july	august	september	october	november	december	TOTAL
FIRE CALL	38	31	48	10	24	12	25	25	24	21	16	25	299
RESCUE CALL	2	1	5	2	1	2	0	2	4	2	3	2	26
EMERGENCY CALL	6	1	0	1	8	0	0	3	2	9	2	5	37
AMBULANCE CALL	0	1	2	0	2	0	1	2	0	1	6	9	24
VIP DUTIES	37	37	45	1	1	2	15	33	32	38	32	33	306
SPECIAL SERVICE	24	19	12	11	2	7	1	6	6	5	10	20	123
TEST CALL	3	3	3	0	1	5	1	2	5	5	7	9	44
INSPECTION OF DANGER PLASE	0	0	0	0	0	0	0	0	0	39	16	40	95
TOTAL	110	93	115	25	39	28	43	73	73	120	92	143	954

prepared by - K.T.S.Fernando

K.P.P.R Nanayakkara
Control Room officer

A.P.J.Preethilal
Station officer
(Communucation)

W.S.R.N Senanayake
Divisional fire officer
(Operation)

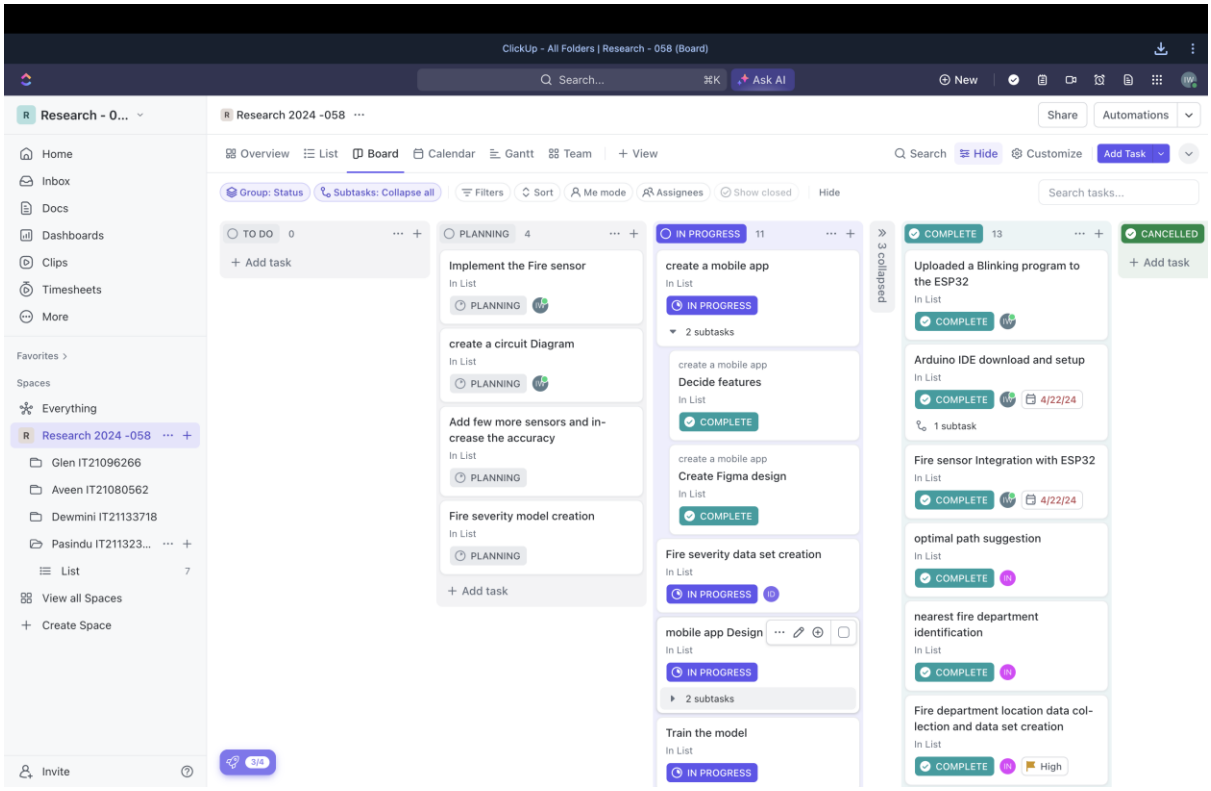
P.D.K.A.Wilson
Chief fire officer

Snapshots from Field Visit

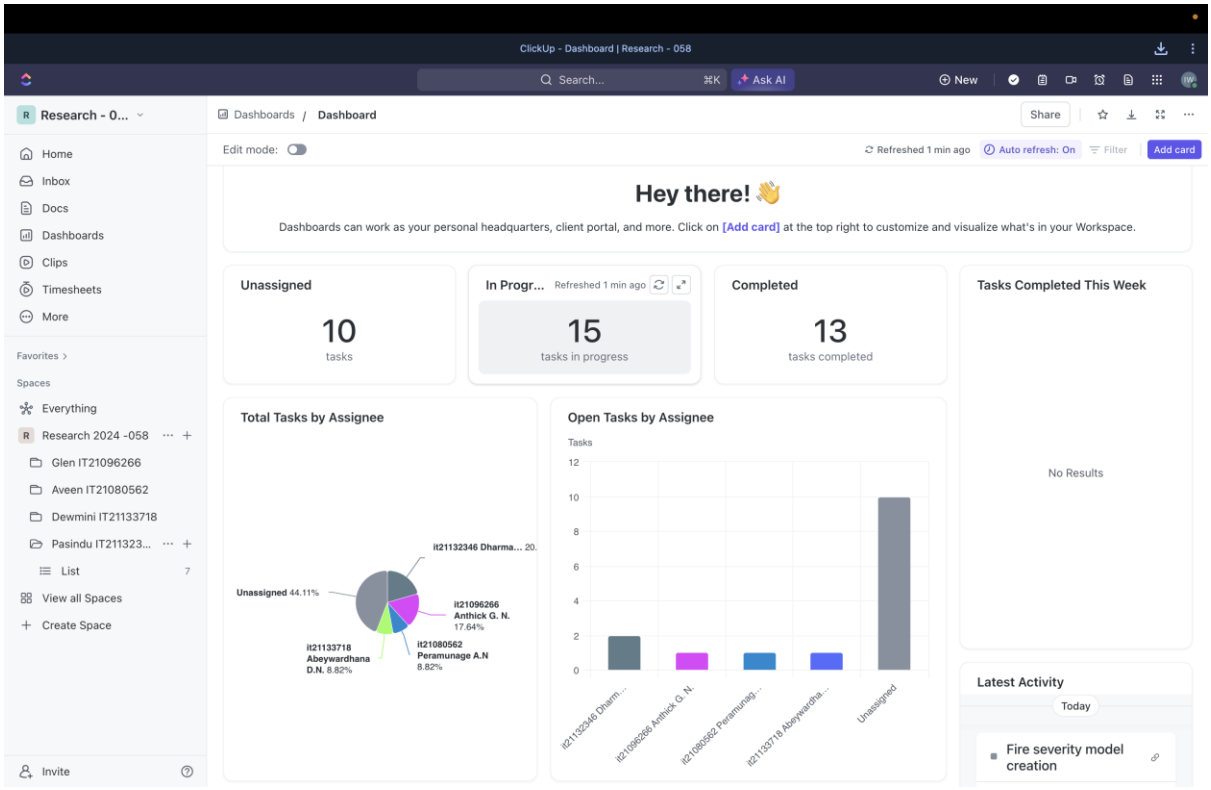

MAN Truck & Bus SE
WMA52WZZ9LL076990
R134a
1.1 kg
DATE


COLOMBO
MUNICIPAL
COUNCIL

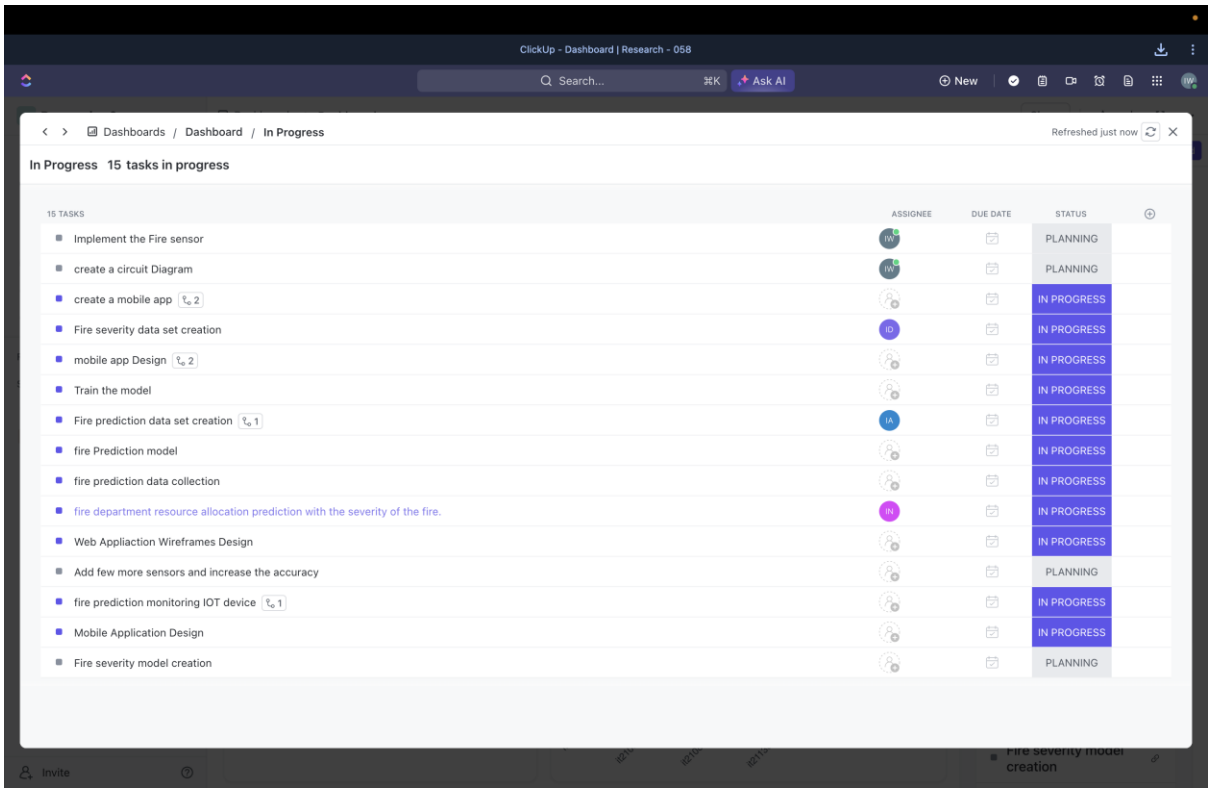
Clickup Tasks Allocation



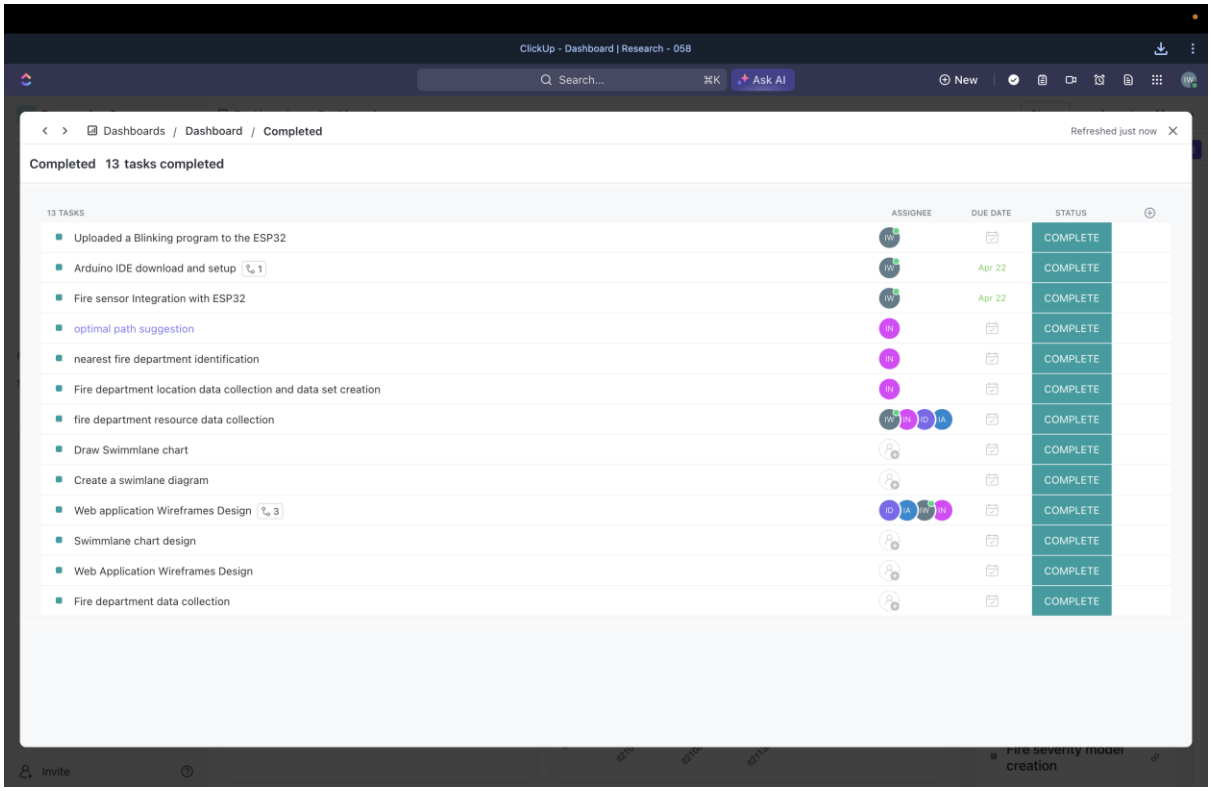
CkickUp Dashboard



In Progress Tasks



Completed Tasks upto PP1



Project Implementation

Data Collection

Collectiong the temprature and RPM data using the themometer and a RPM guge.

Vehicle Safety System for Fire Detection and Prevention

Deep Learning CNN Model

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv1D, Flatten, MaxPooling1D, Dropout, BatchNormalization
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau, LearningRateScheduler
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.regularizers import l2
import numpy as np
```

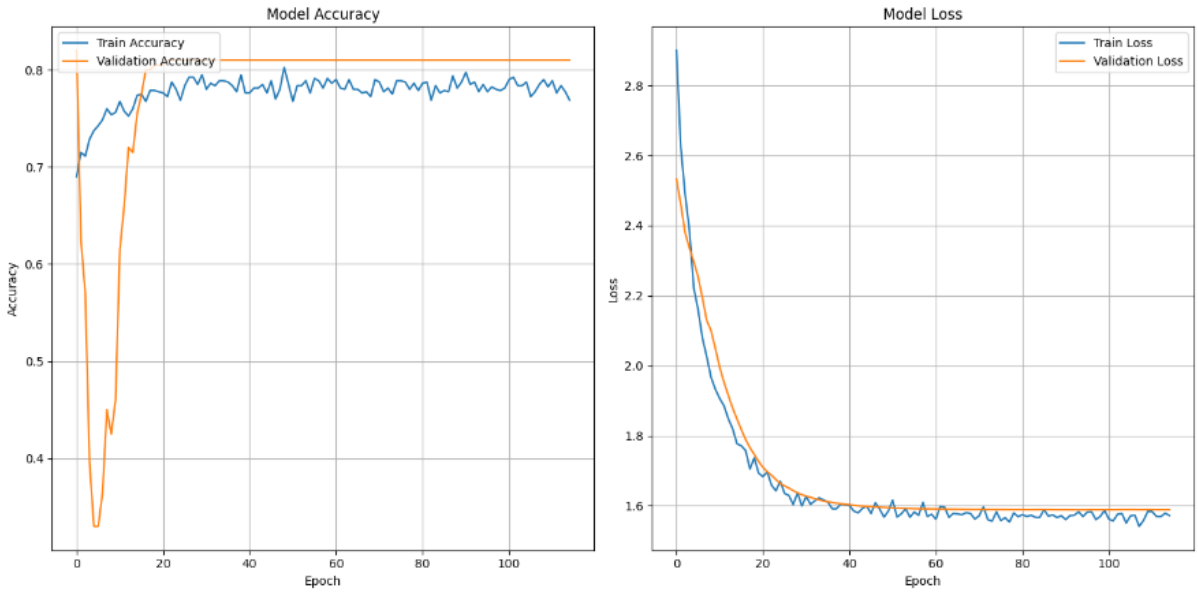
```
# Load the dataset
df = pd.read_csv('sensor_reading_abno.csv')

# Prepare the input features and labels
X = df[['1000_rpm_temprature', '2000_rpm_temprature', '3000_rpm_temprature', '4000_rpm_temprature',
        '5000_rpm_temprature', 'cabin_without_ac_sunny_day', 'rear_without_ac_sunny_day',
        '1000_rpm', '2000_rpm', '3000_rpm', '4000_rpm', '5000_rpm']].values
y = df['label'].apply(lambda x: 0 if x == 'normal' else 1).values
y = to_categorical(y)
```

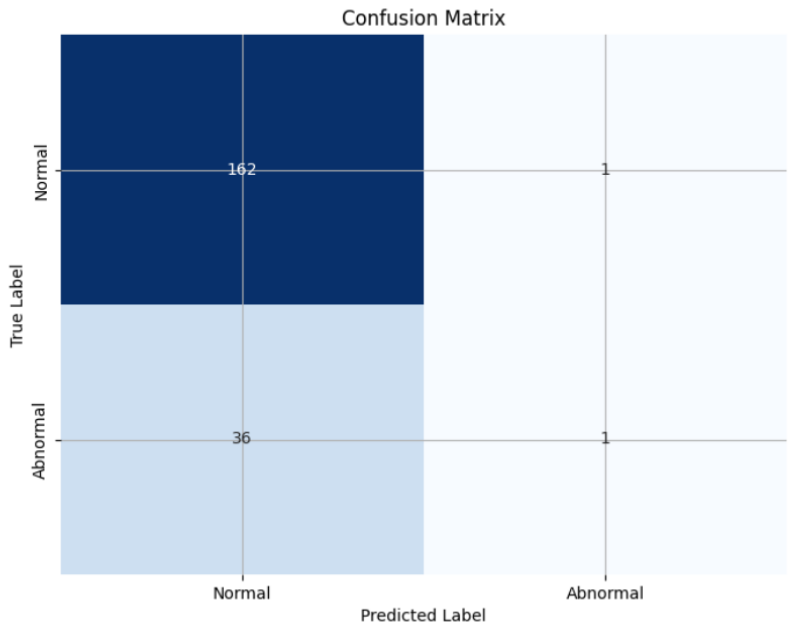
```
# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split( *arrays: X, y, test_size=0.2, random_state=42)

# Normalize features using StandardScaler
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Reshape X to fit the model's expected input
X_train = X_train.reshape(X_train.shape[0], X_train.shape[1], 1)
X_test = X_test.reshape(X_test.shape[0], X_test.shape[1], 1)

# Define the model with additional regularization and layers
model = Sequential([
    Conv1D(filters=64, kernel_size=3, activation='relu', input_shape=(12, 1), kernel_regularizer=l2(0.01)),
    BatchNormalization(),
    MaxPooling1D(pool_size=2),
    Dropout(0.35),
    Conv1D(filters=128, kernel_size=3, activation='relu', kernel_regularizer=l2(0.01)),
```

```python
# Define the model with additional regularization and layers
model = Sequential([
    Conv1D(filters=64, kernel_size=3, activation='relu', input_shape=(12, 1), kernel_regularizer=l2(0.01)),
    BatchNormalization(),
    MaxPooling1D(pool_size=2),
    Dropout(0.35),
    Conv1D(filters=128, kernel_size=3, activation='relu', kernel_regularizer=l2(0.01)),
    BatchNormalization(),
    MaxPooling1D(pool_size=2),
    Dropout(0.35),
    Flatten(),
    Dense( units: 100, activation='relu', kernel_regularizer=l2(0.01)),
    Dropout(0.5),
    Dense( units: 2, activation='softmax')
])

# Compile the model
model.compile(optimizer=Adam(learning_rate=0.001), loss='categorical_crossentropy', metrics=['accuracy'])
```

```python
# Compile the model
model.compile(optimizer=Adam(learning_rate=0.001), loss='categorical_crossentropy', metrics=['accuracy'])

# Define callbacks
early_stopping = EarlyStopping(monitor='val_loss', patience=20, restore_best_weights=True)
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=10, min_lr=0.0001, verbose=1)
lr_scheduler = LearningRateScheduler(lambda epoch: 0.001 * np.exp(-0.1 * epoch), verbose=1)

# Train the model
history = model.fit(X_train, y_train, epochs=150, batch_size=32, validation_data=(X_test, y_test), verbose=2,
                    callbacks=[early_stopping, reduce_lr, lr_scheduler])

# Save the model
model.save('enhanced_sensor_model_v2.h5')
```

```python
# Visualization of the training process
plt.figure(figsize=(14, 7))
plt.subplot( *args: 1, 2, 1)
plt.plot( *args: history.history['accuracy'], label='Train Accuracy')
plt.plot( *args: history.history['val_accuracy'], label='Validation Accuracy')
plt.title('Model Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.grid(True)
plt.legend(loc='upper left')

plt.subplot( *args: 1, 2, 2)
plt.plot( *args: history.history['loss'], label='Train Loss')
plt.plot( *args: history.history['val_loss'], label='Validation Loss')
plt.title('Model Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.grid(True)
plt.legend(loc='upper right')
plt.tight_layout()
plt.show()

# Evaluate the model on test data
test_loss, test_accuracy = model.evaluate(X_test, y_test, verbose=2)
print(f"Test Loss: {test_loss}, Test Accuracy: {test_accuracy}")
```






```
Epoch 115: LearningRateScheduler setting learning rate to 1.1195484842590946-08.
Epoch 115/150
25/25 - 0s - loss: 1.5723 - accuracy: 0.7688 - val_loss: 1.5887 - val_accuracy: 0.8100 - lr: 1.1195e-08 - 96ms/epoch - 4ms/step
7/7 - 0s - loss: 1.5884 - accuracy: 0.8100 - 39ms/epoch - 6ms/step
Test Loss: 1.5884194374084473, Test Accuracy: 0.8100000023841858
```

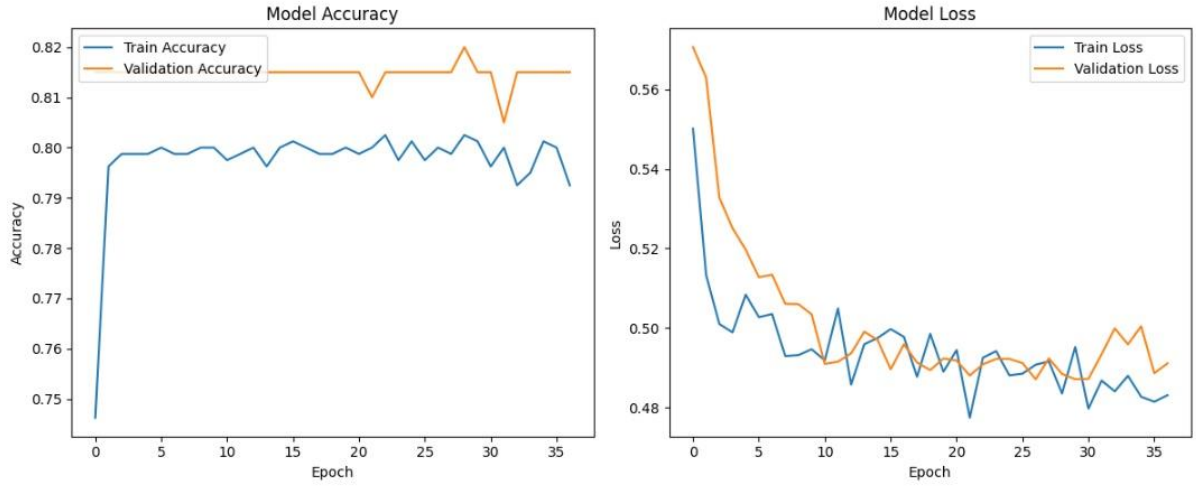
Deep Learning RNN Model(LSTM)

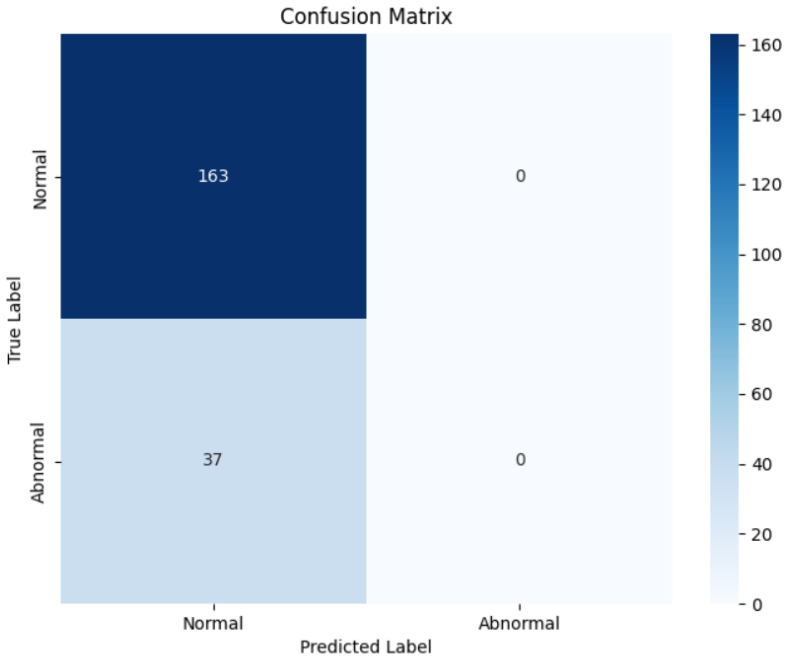
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout, BatchNormalization
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
from tensorflow.keras.optimizers import Adam

# Load the dataset
df = pd.read_csv('sensor_reading_abno.csv')

# Prepare the input features and labels
features = ['1000_rpm_temprature', '1000_rpm', '2000_rpm_temprature', '3000_rpm_temprature',
            '4000_rpm_temprature', '5000_rpm_temprature', 'cabin_without_ac_sunny_day',
            'rear_without_ac_sunny_day', '2000_rpm', '3000_rpm', '4000_rpm', '5000_rpm']
X = df[features].values
y = df['label'].apply(lambda x: 0 if x == 'normal' else 1).values
y = to_categorical(y)

# Normalize features using StandardScaler
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_scaled = X_scaled.reshape(X_scaled.shape[0], X_scaled.shape[1], 1)

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split( *arrays: X_scaled, y, test_size=0.2, random_state=42)
```

```python
# Define the LSTM model
model = Sequential([
    LSTM( units: 64, return_sequences=True, input_shape=(12, 1)),
    Dropout(0.3),
    BatchNormalization(),
    LSTM( units: 64, return_sequences=False),
    Dropout(0.3),
    Dense( units: 100, activation='relu'),
    Dropout(0.3),
    Dense( units: 2, activation='softmax')
])

# Compile the model with Adam optimizer and categorical crossentropy loss
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Setup callbacks for early stopping (to prevent overfitting) and best model checkpointing
early_stopping = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)
checkpoint = ModelCheckpoint( filepath: 'best_model.h5', save_best_only=True, monitor='val_loss', mode='min')

# Train the model
history = model.fit(
    X_train, y_train,
    epochs=200,
    batch_size=32,
    validation_data=(X_test, y_test),
    callbacks=[early_stopping, checkpoint],
    verbose=2
)
```

```python
# Load the best saved model
model.load_weights('best_model.h5')

# Predictions
y_pred = model.predict(X_test)
y_pred_classes = np.argmax(y_pred, axis=1)
y_true = np.argmax(y_test, axis=1)

# Confusion Matrix
cm = confusion_matrix(y_true, y_pred_classes)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=['Normal', 'Abnormal'], yticklabels=['Normal', 'Abnormal'])
plt.title('Confusion Matrix')
plt.ylabel('True Label')
plt.xlabel('Predicted Label')
plt.show()

# Classification Report and Accuracy
print("Accuracy: {:.2f}%".format(accuracy_score(y_true, y_pred_classes) * 100))
print("\nClassification Report:\n", classification_report(y_true, y_pred_classes))
```

```python
# Plot Training and Validation Accuracy
plt.figure(figsize=(12, 5))
plt.subplot( *args: 1, 2, 1)
plt.plot( *args: history.history['accuracy'], label='Train Accuracy')
plt.plot( *args: history.history['val_accuracy'], label='Validation Accuracy')
plt.title('Model Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend(loc='upper left')

# Plot Training and Validation Loss
plt.subplot( *args: 1, 2, 2)
plt.plot( *args: history.history['loss'], label='Train Loss')
plt.plot( *args: history.history['val_loss'], label='Validation Loss')
plt.title('Model Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend(loc='upper right')
plt.tight_layout()
plt.show()
```





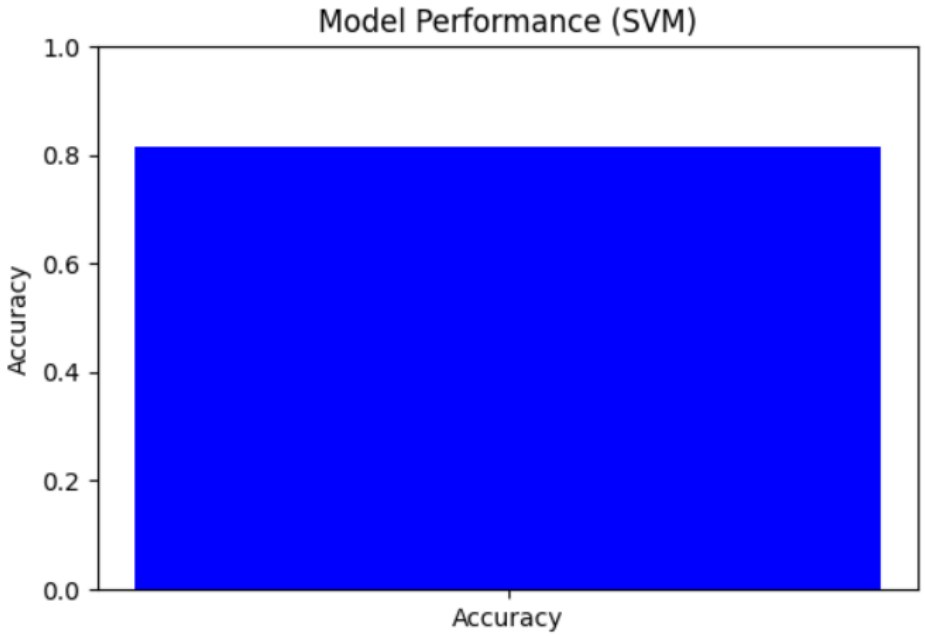

```
25/25 - 0s - loss: 0.4831 - accuracy: 0.7925 - val_loss: 0.4911 - val_accuracy: 0.8150 - 213ms/epoch - 9ms/step
7/7 [==============================] - 1s 3ms/step
Accuracy: 81.50%
```

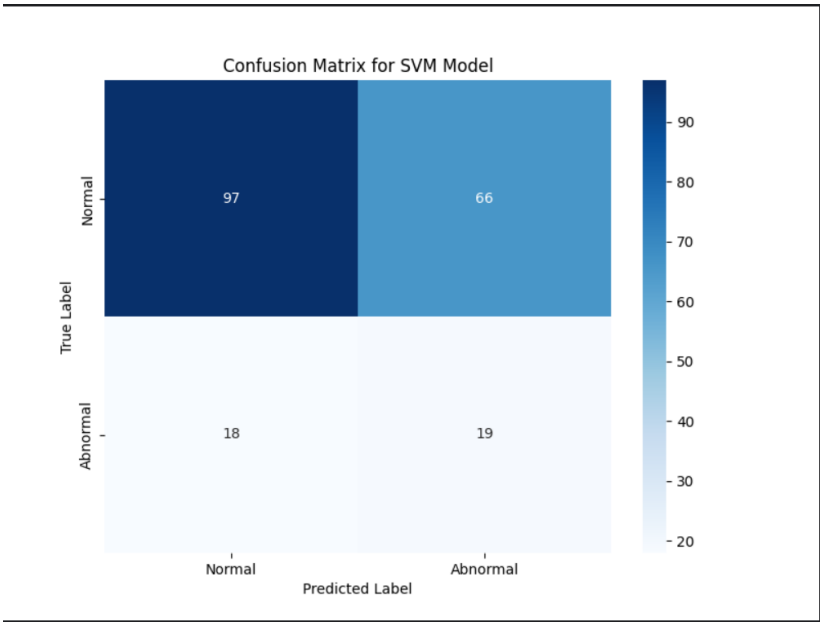
```
Classification Report:
              precision    recall  f1-score   support

           0       0.81      1.00      0.90       163
           1       0.00      0.00      0.00        37

    accuracy                           0.81       200
   macro avg       0.41      0.50      0.45       200
weighted avg       0.66      0.81      0.73       200
```

Deep learning SVM Model

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
from sklearn.svm import SVC

# Load the dataset
df = pd.read_csv('sensor_reading_abno.csv')

# Prepare the input features and labels
features = ['1000_rpm_temprature', '1000_rpm', '2000_rpm_temprature', '3000_rpm_temprature',
            '4000_rpm_temprature', '5000_rpm_temprature', 'cabin_without_ac_sunny_day',
            'rear_without_ac_sunny_day', '2000_rpm', '3000_rpm', '4000_rpm', '5000_rpm']
X = df[features].values
y = df['label'].apply(lambda x: 0 if x == 'normal' else 1).values

# Normalize features using StandardScaler
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split( *arrays: X_scaled, y, test_size=0.2, random_state=42)
```

```python
# Build SVM model
svm_model = SVC(class_weight='balanced')  # Adjusting class weights for handling imbalanced data
svm_model.fit(X_train, y_train)

# Predict labels for test set
y_pred = svm_model.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print("SVM Accuracy:", accuracy)

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt="d", cmap='Blues', xticklabels=['Normal', 'Abnormal'], yticklabels=['Normal', 'Abnormal'])
plt.title('Confusion Matrix for SVM Model')
plt.ylabel('True Label')
plt.xlabel('Predicted Label')
plt.show()

# Classification Report with zero_division=1
print("\nClassification Report:\n", classification_report(y_test, y_pred, zero_division=1))
```

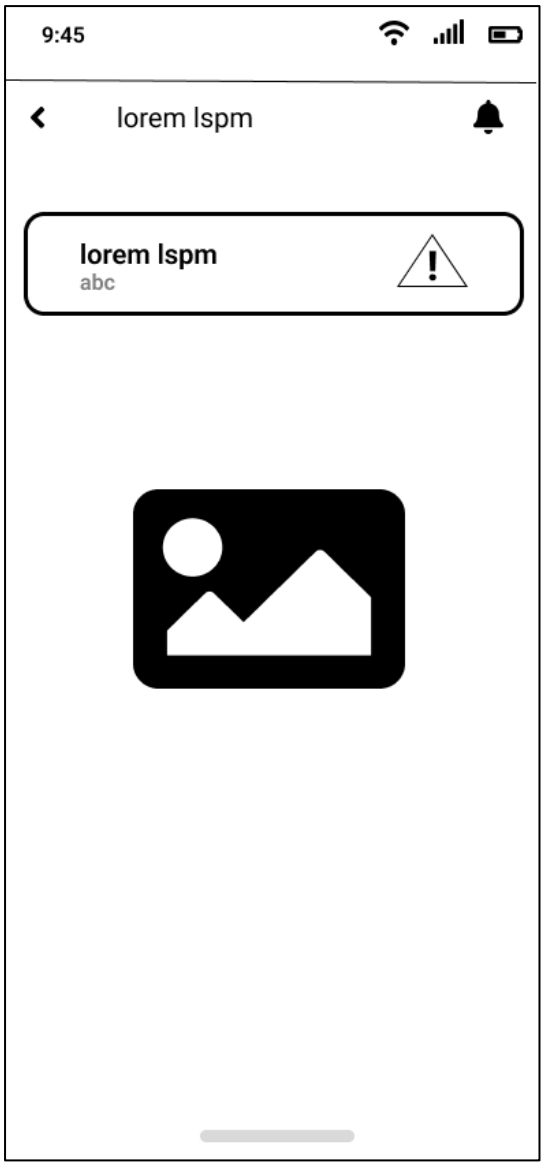

Model Performance (SVM)
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
Accuracy

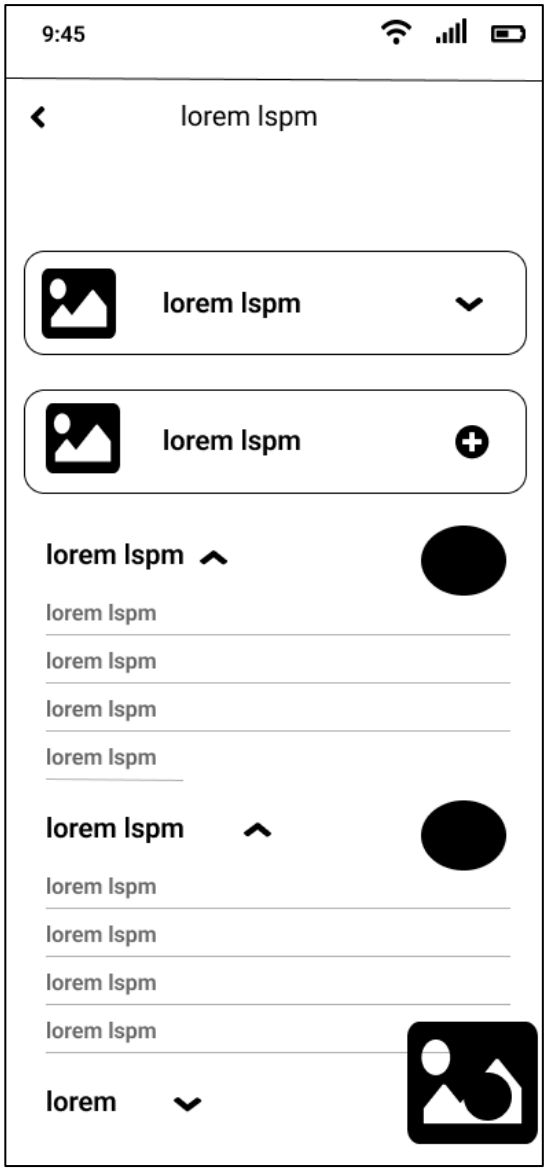

Confusion Matrix for SVM Model
Normal
Abnormal
True Label
97
66
18
19
90
80
70
60
50
40
30
20
Normal
Abnormal
Predicted Label

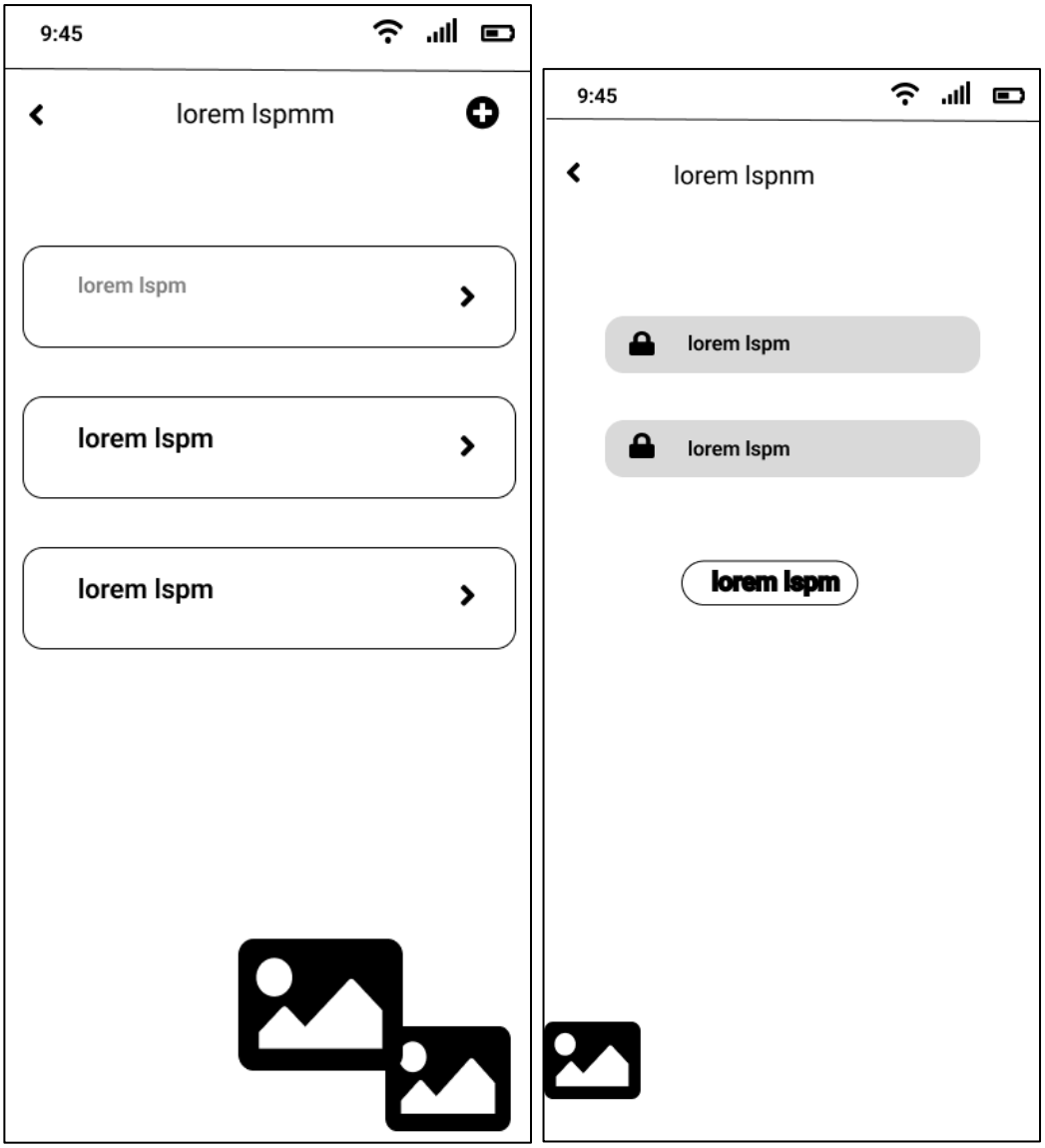
```
SVM Accuracy: 0.58

Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.60      0.70       163
           1       0.22      0.51      0.31        37

    accuracy                           0.58       200
   macro avg       0.53      0.55      0.50       200
weighted avg       0.73      0.58      0.63       200
```

Used Algorithm	Overall Accuracy
CNN Algoritham	82.0%
Support Vector Machine(SVM)	81.50%
RNN Model(LSTM)	81.50%

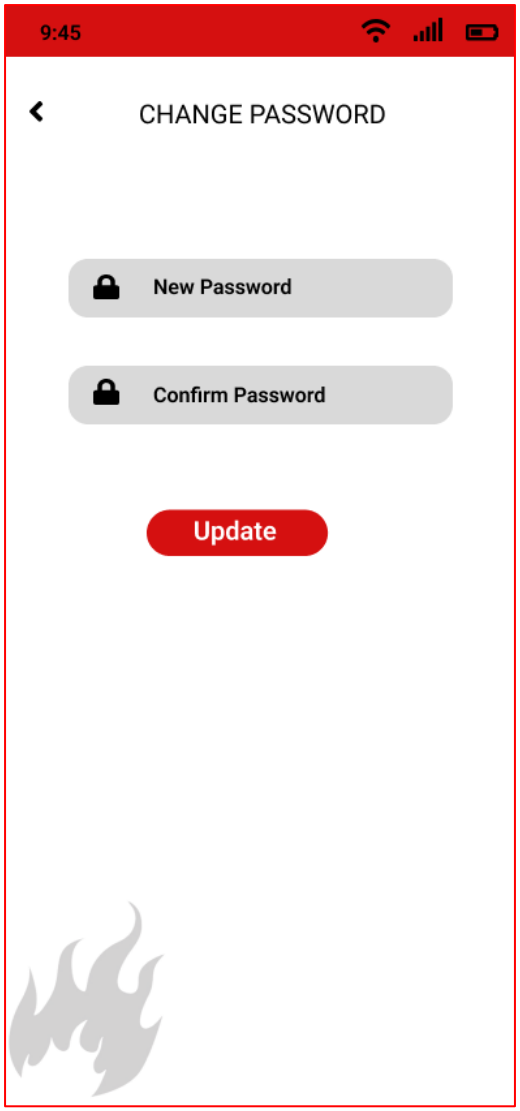
Wireframes

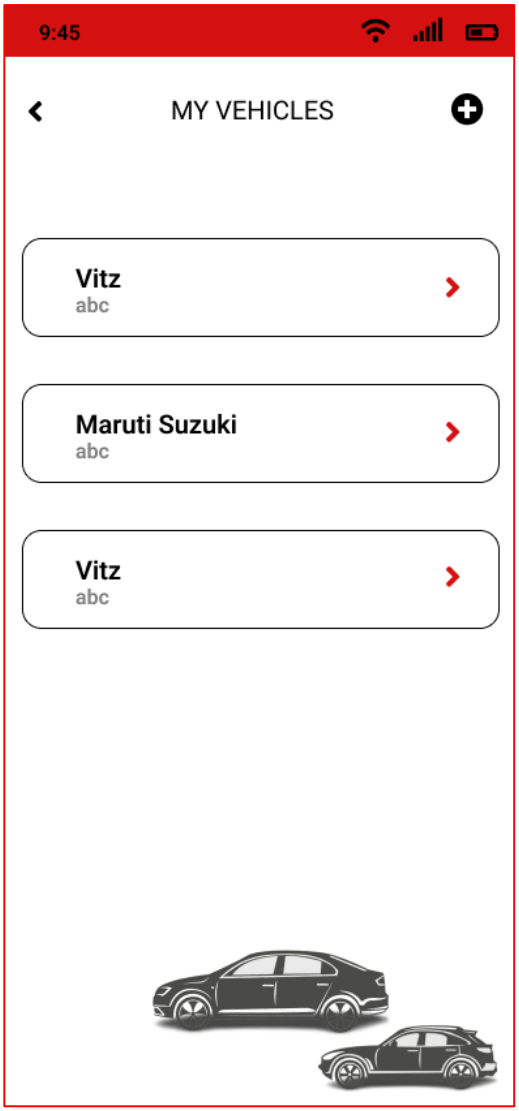

9:45
lorem lspm
lorem lspm
abc

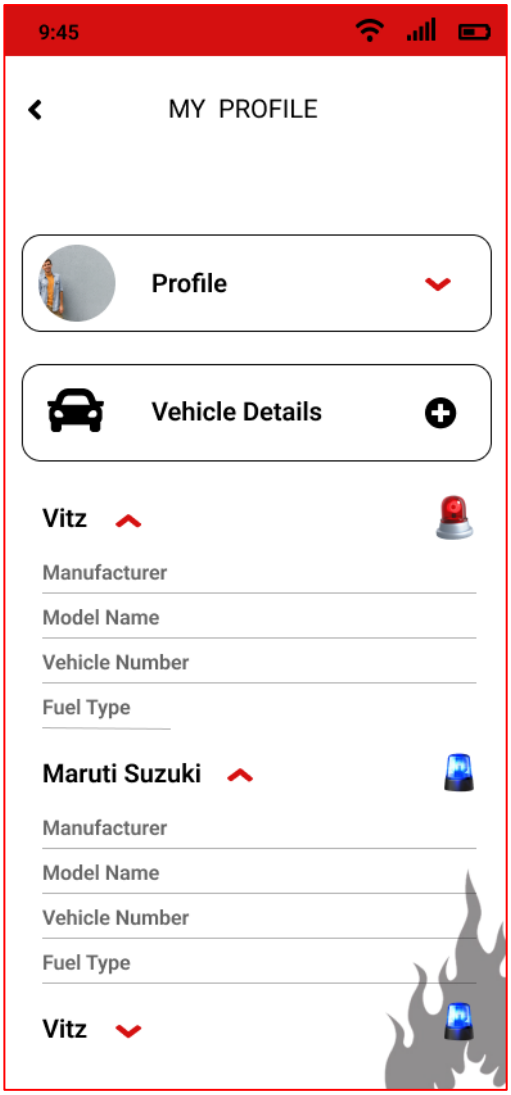

9:45
lorem lspm
lorem lspm
lorem lspm
lorem lspm
lorem lspm
lorem lspm
lorem lspm
lorem lspm
lorem lspm
lorem lspm
lorem lspm
lorem lspm
lorem


9:45
lorem lspmm
lorem lspm
lorem lspm
lorem lspm
9:45
lorem lspnm
lorem lspm
lorem lspm
lorem lspm

Mobile Application UI

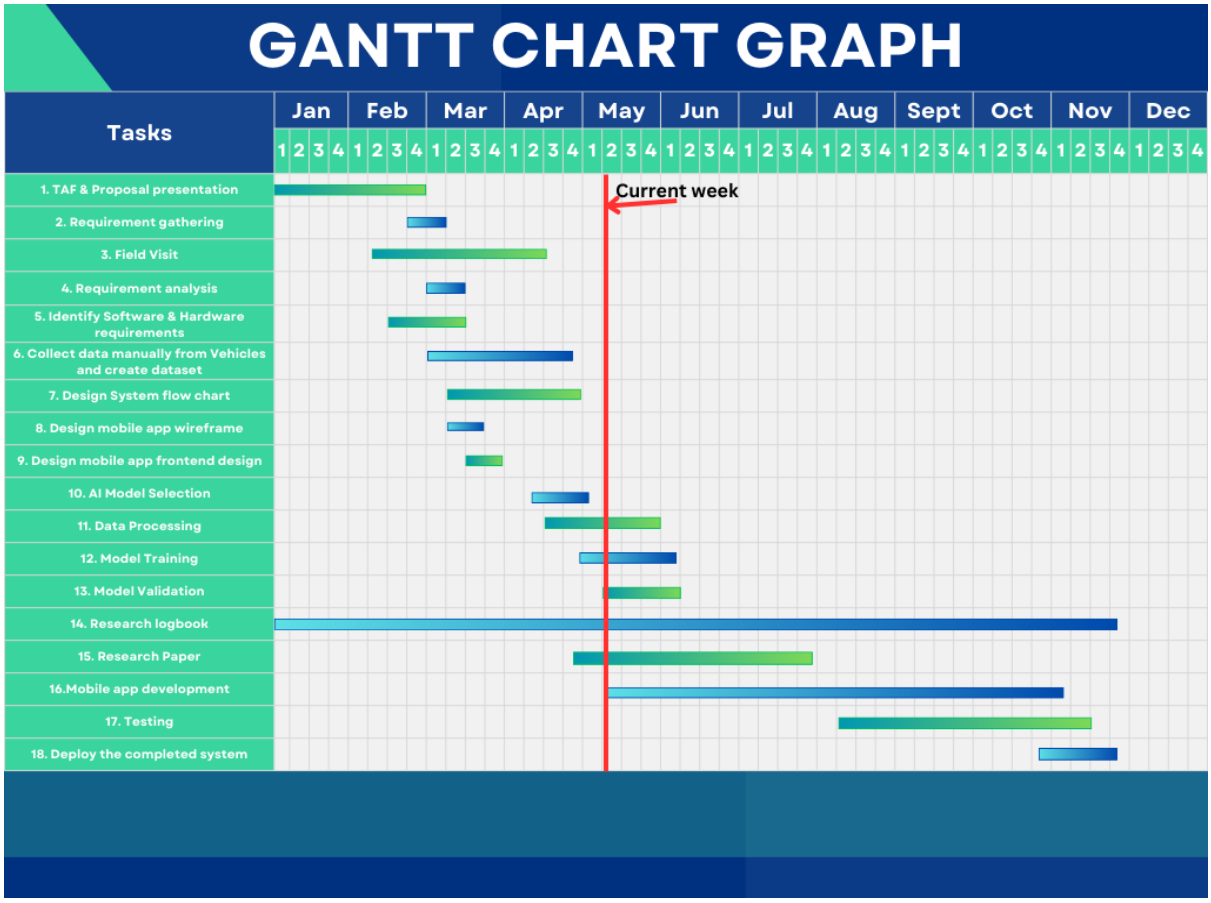

9:45
CHANGE PASSWORD
New Password
Confirm Password
Update


9:45
MY VEHICLES
Vitz
abc
Maruti Suzuki
abc
Vitz
abc


9:45
MY PROFILE
Profile
Vehicle Details
Vitz
Manufacturer
Model Name
Vehicle Number
Fuel Type
Maruti Suzuki
Manufacturer
Model Name
Vehicle Number
Fuel Type
Vitz


9:45
FIRE ALERT RECEIVED
Vitz
abc

Gantt Chart



Work Breakdown Structure