PlantPal

SMART ASSISTANCE FOR THE FLORICULTURE INDUSTRY

2023-133

SLIIT
FACULTY OF COMPUTING

MEET OUR TEAM

IT19169736

Gamage M.G.U.D.

IT20017088

Prabhashi P.A.N.

IT20005726

Liyanage S.R.

IT19994406

Basnayake N.S.N.

SLIIT
FACULTY OF COMPUTING

2023-133

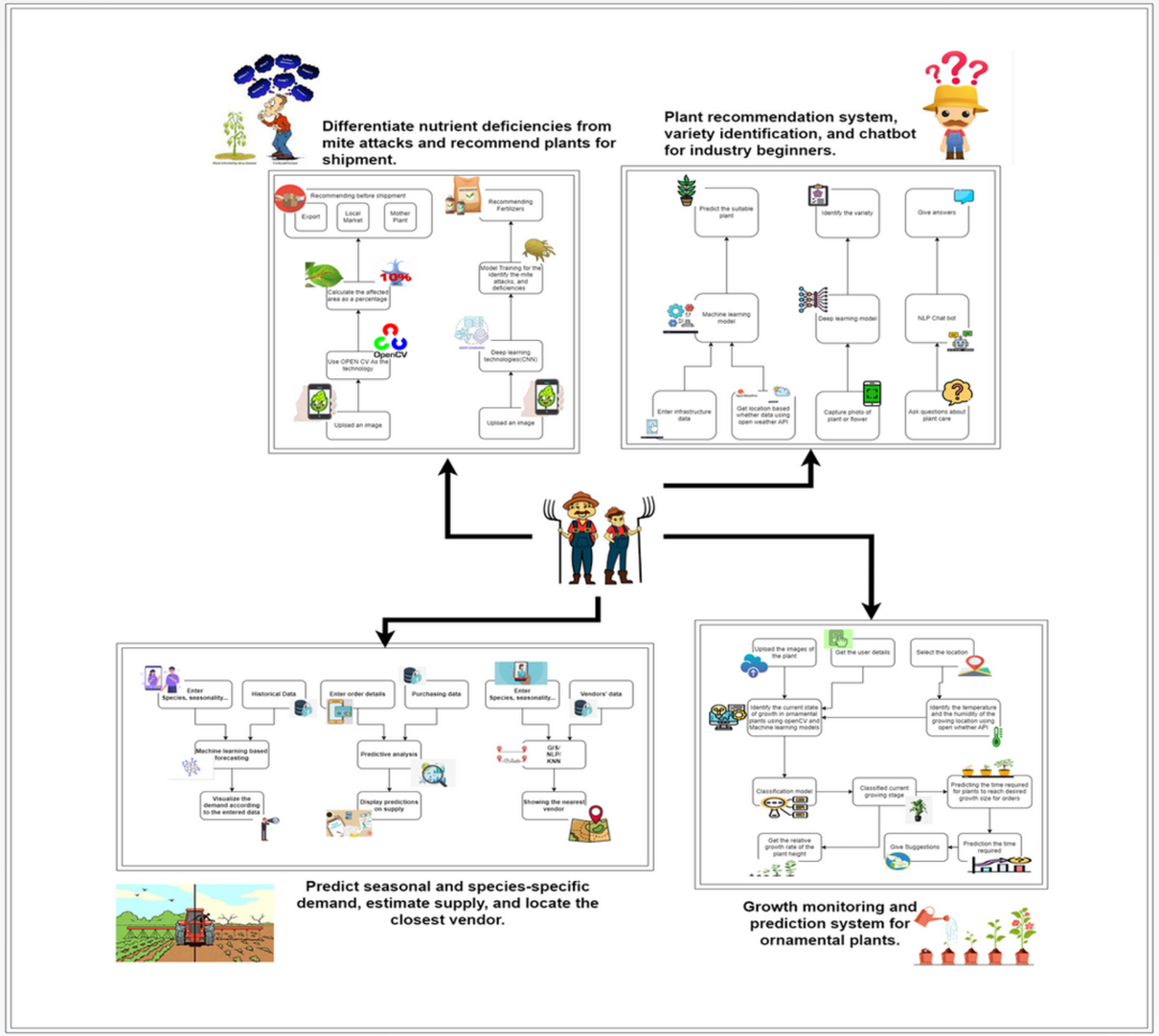
INTRODUCTION

- Revolutionize Sri Lanka's Floriculture with Smart Tech
- Empower Beginners & Interns with PlantPal's Expertise
- Your Companion on iOS & Android
- Remote Access with Separated Backend & Frontends
- Seamless Experience: Web & Mobile Ready

SLIIT
FACULTY OF COMPUTING

2023-133

OVERALL SYSTEM ARCITECTURE

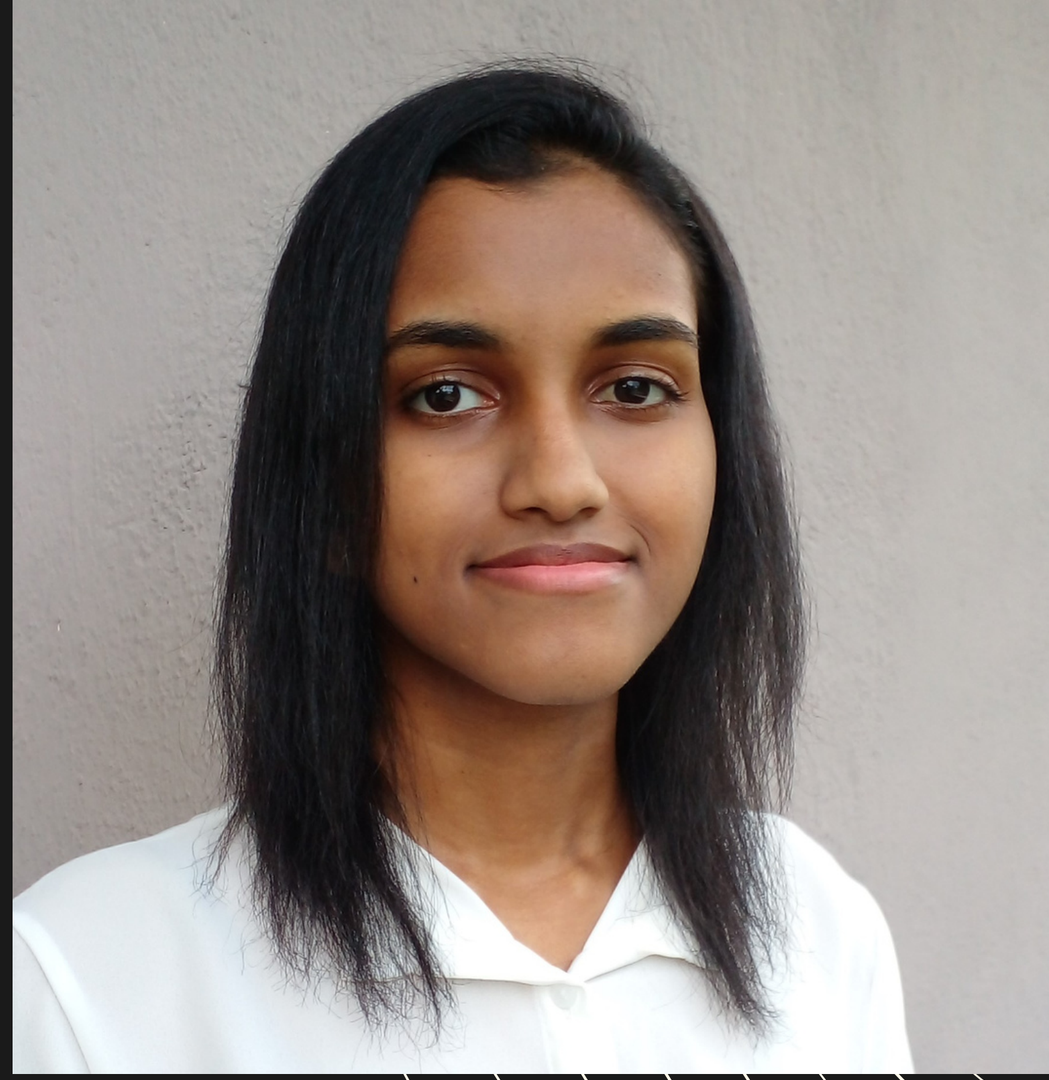


SLIIT
FACULTY OF COMPUTING

2023-133

IT19169736
Gamage M.G.U.D.

CUT FLOWER CROP SELECTION & PHILODENDRON VARIETY IDENTIFICATION SYSTEM FOR FLORICULTURE BEGINNERS



SLIIT
FACULTY OF COMPUTING

Specialization : Data Science

RESEARCH PROBLEM

- Inexperienced flower growers facing losses in resources, money, and time due to the lack of guidance in selecting suitable cut flower crops based on their local weather conditions, available resources, and yield goals

SLIIT
FACULTY OF COMPUTING

RESEARCH PROBLEM

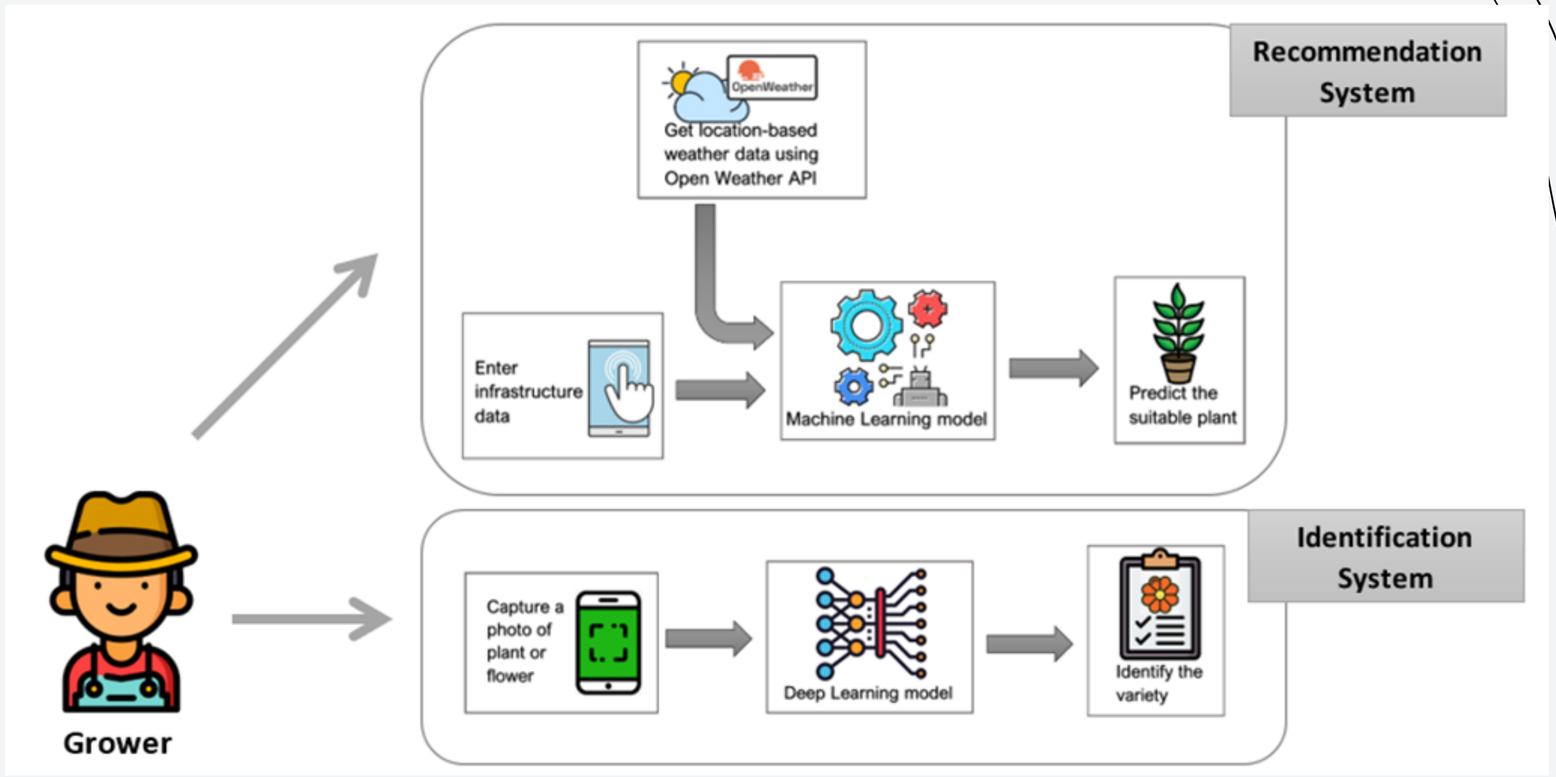
- Inexperienced growers struggle with the task of correctly recognizing 'Philodendron' varieties, which limits the production of attractive leaves for export and tissue culture.

PREVIOUS STUDIES VS THIS STUDY

- Utilizing IoT Sensors and Machine Learning for Crop Selection
- Anthurium Variety Identification on Android platform

- Affordable Cutflower Crop Selection using **OpenWeather API** & ML - **Cutflower Advisor**
- Philodendron Variety Identification, a First in Sri Lanka, on **Android and iOS** - **Philovariety Finder**

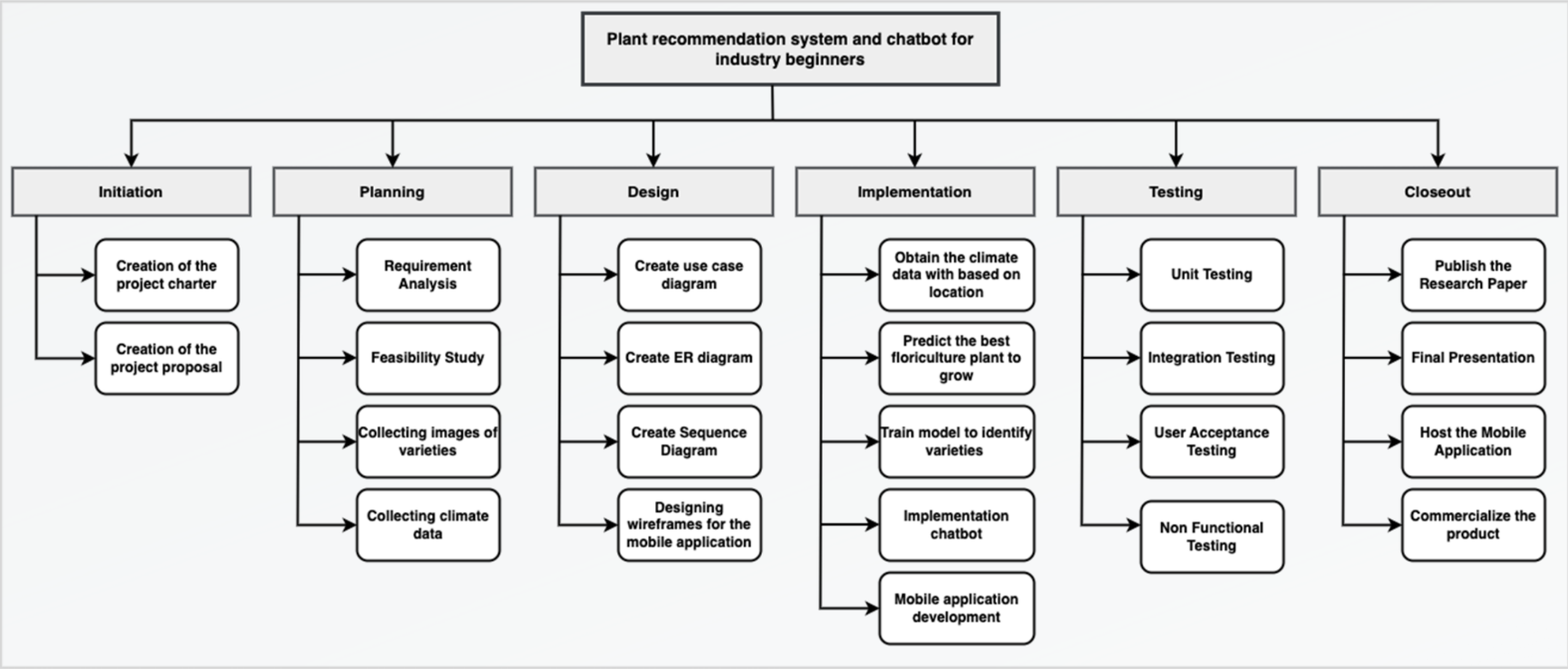
INDIVIDUAL SYSTEM DIAGRAM



SLIIT
FACULTY OF COMPUTING

IT19169736 | Gamage M.G.U.D. | 2023-133

WORK BREAKDOWN



SLIIT
FACULTY OF COMPUTING

IT19169736 | Gamage M.G.U.D. | 2023-133

RESEARCH PROGRESS

Cutflower Advisor

PP 1

- Data collection & preprocessing
- Exploratory Data Analysis
- ML model comparison
- Model training
- Testing & evaluation
- Obtaining weather data - Open Weather API
- Prediction
- Designing UI

PP 2

- Feature selection
- Model justification
- Learning curves
- Hyper parameter tuning
- Improving accuracy
- Developing Flask API
- React Native app

SLIIT
FACULTY OF COMPUTING

IT19169736 | Gamage M.G.U.D. | 2023-133

RESEARCH PROGRESS

PhiloVariety Finder

PP 1

- Data collection

PP 2

- Data preprocessing, Exploratory Data Analysis
- Feature extraction, Adding layers
- Data augmentation
- Building a custom CNN
- Training & testing selected model
- Hyper parameter tuning
- Training with best model and testing
- Comparing learning curves
- Model evaluation, Predicting on a test set
- Binary classifier for validation
- Prediction
- Developing Flask API, React Native app

IMPLEMENTATION - CUTFLOWER ADVISOR
INPUTS TO THE SYSTEM

Feature	Importance in floriculture
Temperature,Relative Humidity (using Open Weather API)	Environmental factors
Land size	Crop Diversity
Initial plant count	Crop spacing
Initial budget	Long-Term Planning
Monthly resources	Crop Health
Expected flower yield	To assess whether the yield can be achieved

IMPLEMENTATION - CUTFLOWER ADVISOR

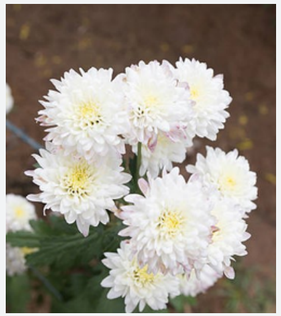
OUTPUT OF THE SYSTEM

Predicts the cut flower crop from 8 classes

Orchid

Rose

Lily

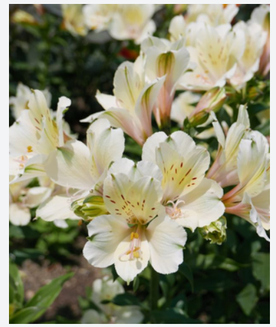


Chrysanthemum

Anthurium

Gerbera

Carnation



Alstromeria

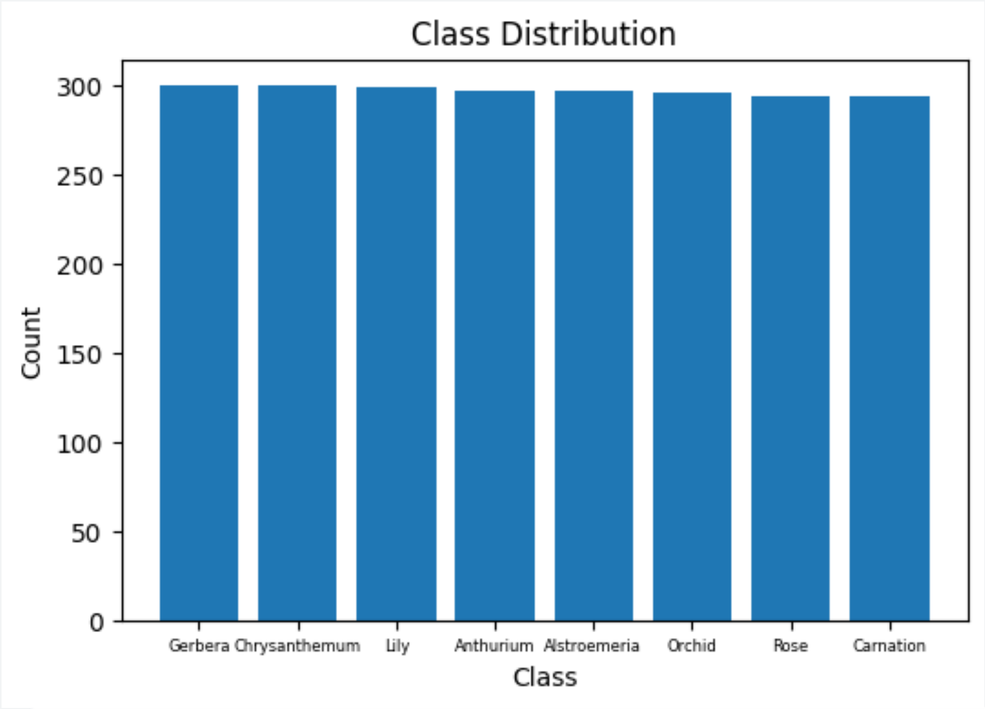
SLIIT FACULTY OF COMPUTING

IT19169736 | Gamage M.G.U.D. | 2023-133

IMPLEMENTATION - CUTFLOWER ADVISOR METHODOLOGY OF THE SYSTEM

1. Data collection
2. Data preprocessing
3. Exploratory Data Analysis

Temperature	float64
Relative_Humidity	float64
Water_liters_per_yr	int64
Land_size_in_sqft	float64
No_of_plants	float64
Nethouse_land_preperation_cost	float64
Planting_cost	float64
Labour_cost	float64
Maintaining_cost	int64
Flower_yield_per_yr	float64
Crop_name	object



Class distribution of the dataset

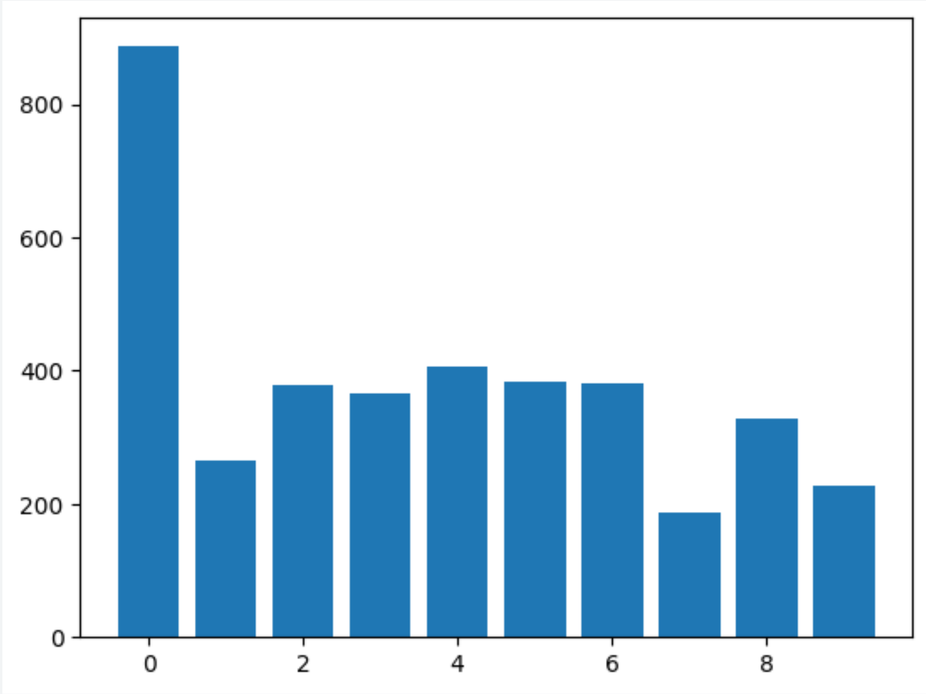
SLIIT FACULTY OF COMPUTING

IT19169736 | Gamage M.G.U.D. | 2023-133

IMPLEMENTATION - CUTFLOWER ADVISOR METHODOLOGY OF THE SYSTEM

4. Feature Selection

- **ANOVA F-value** for each feature is calculated
- The significance in relation to the target variable
- **Top 9** features are selected



Feature importance graph

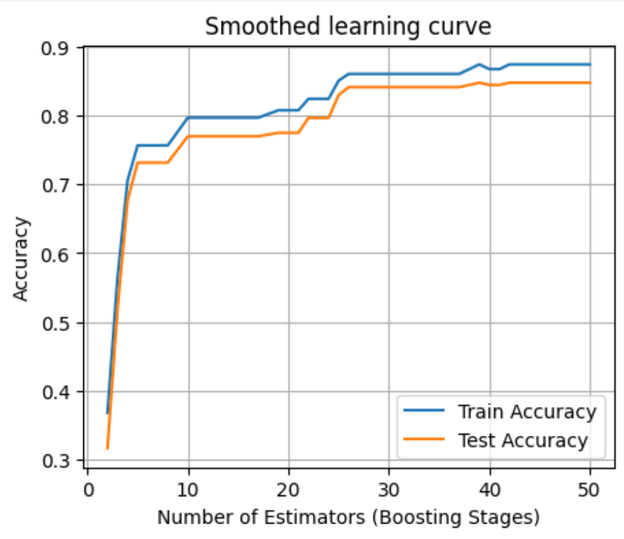
IMPLEMENTATION - CUTFLOWER ADVISOR METHODOLOGY OF THE SYSTEM

5. Model justification

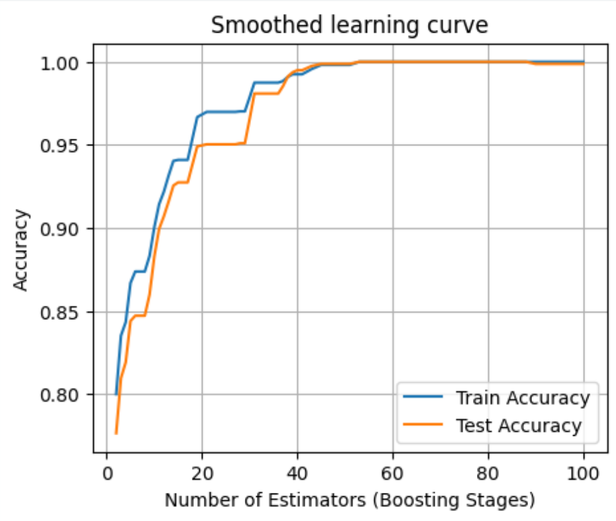
- 3 models are selected for the Multiclass Classification problem
- **Decision Tree -** Good starting point for understanding the data's structure
- **Naive Bayes -** Perform well even with relatively small datasets
- **Gradient Boosting -** Ensemble method that combines the predictions of multiple weak learners
- **Using k-fold cross-validation to evaluate the performance,**
- **Gradient Boosting is selected**

IMPLEMENTATION - CUTFLOWER ADVISOR METHODOLOGY OF THE SYSTEM

6. Hyperparameter Tuning



Before

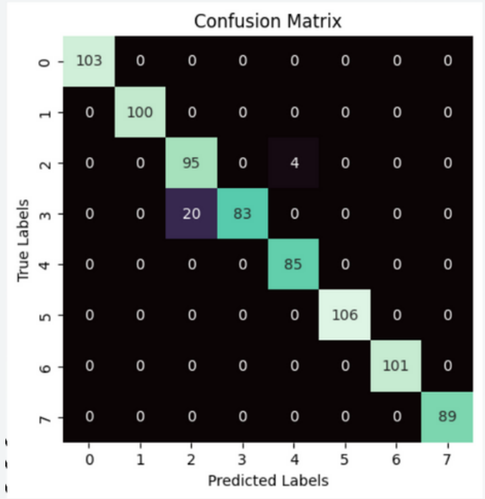


After

IMPLEMENTATION - CUTFLOWER ADVISOR METHODOLOGY OF THE SYSTEM

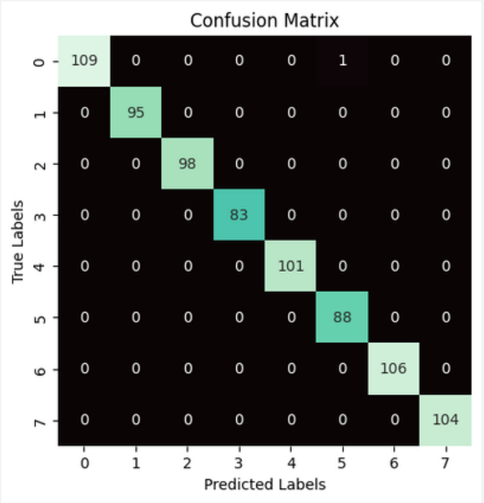
7. Improving Testing Accuracy

96.95%



Before

99.87%



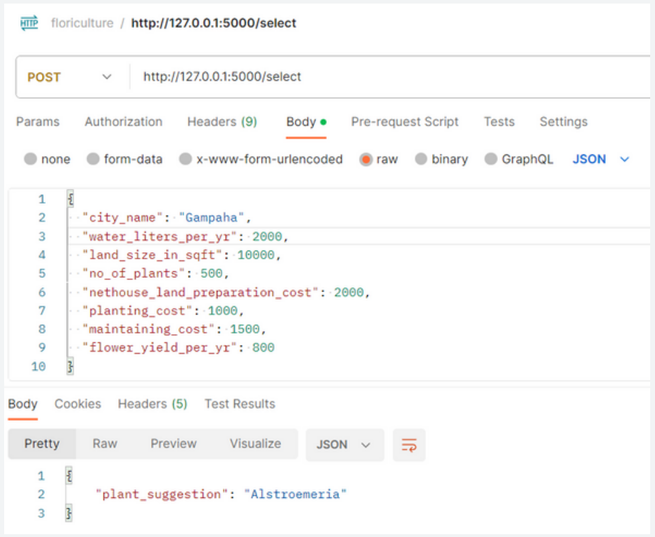
After

SLIIT FACULTY OF COMPUTING

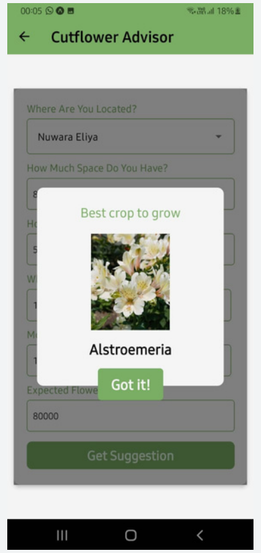
IT19169736 | Gamage M.G.U.D. | 2023-133

IMPLEMENTATION - CUTFLOWER ADVISOR METHODOLOGY OF THE SYSTEM

8. Flask API & React Native mobile app



Testing Flask API on Postman



Output in React Native app

SLIIT FACULTY OF COMPUTING

IT19169736 | Gamage M.G.U.D. | 2023-133

IMPLEMENTATION - PHILOVARIETY FINDER

INPUT TO THE SYSTEM

Camera or Gallery

Resize & Rescale

input

Image of a Philodendron

IMPLEMENTATION - PHILOVARIETY FINDER

OUTPUT OF THE SYSTEM

Input → **Binary classifier** → Not a Philo within scope → Alert user

Binary classifier → A Philo within scope → **Variety identifier** → Classify into 6 classes

Black cardinal

Sunred

Congo green

Majesty

Moonlight

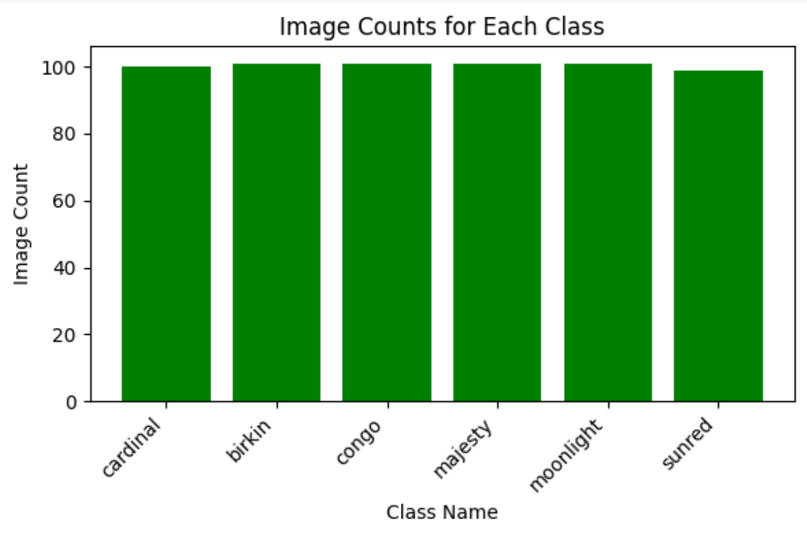
Birkin

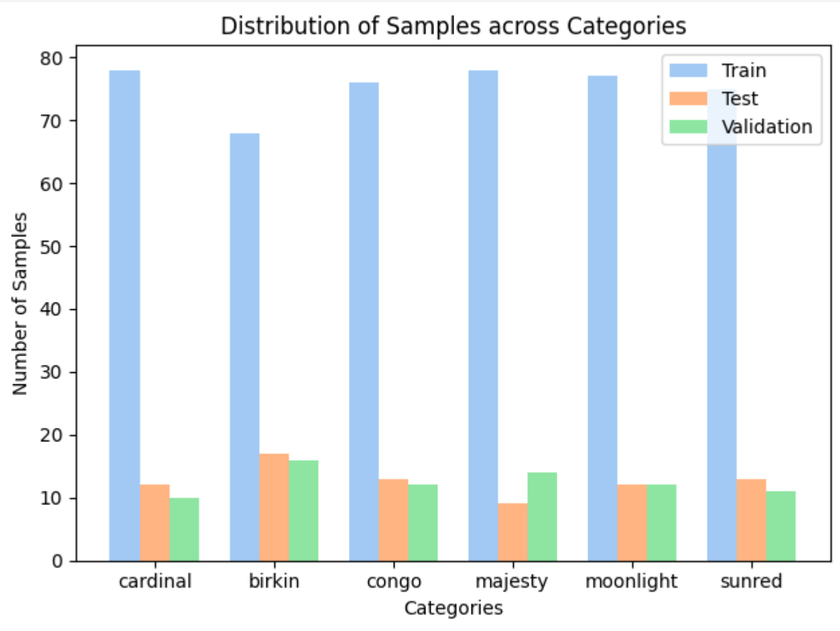
SLIIT FACULTY OF COMPUTING

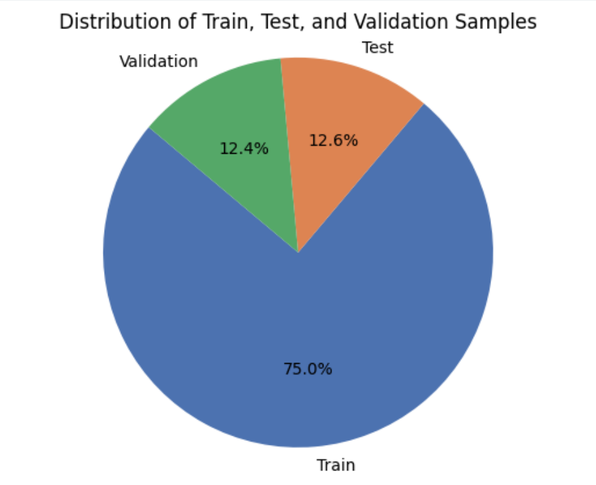
IT19169736 | Gamage M.G.U.D. | 2023-133

IMPLEMENTATION - PHILOVARIETY FINDER METHODOLOGY OF THE SYSTEM

1. Data preprocessing & EDA







SLIIT
FACULTY OF COMPUTING

IT19169736 | Gamage M.G.U.D. | 2023-133

IMPLEMENTATION - PHILOVARIETY FINDER METHODOLOGY OF THE SYSTEM

2. Transfer Learning model vs Custom model

- Pre-trained **MobileNetV2** model as a feature extractor
- Use of the features extracted, as input for additional layers for the Image classification task
- On device real-time inference
- Capable with limited computational resources
- **MobileNetV2** is selected as the best model

Testing accuracy

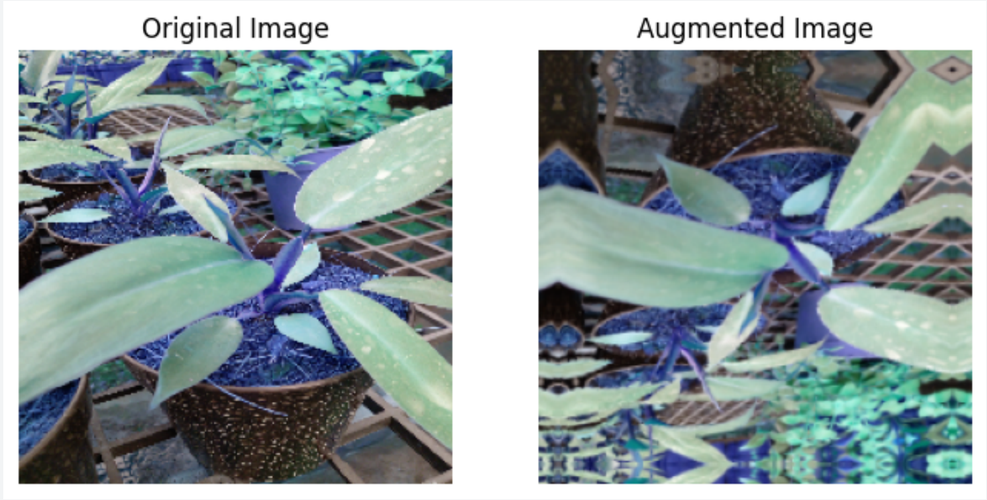
MobileNetV2 : 76.32%

Custom model : 68.42%

IMPLEMENTATION - PHILOVARIETY FINDER METHODOLOGY OF THE SYSTEM

3. Techniques to reduce overfitting

- Data augmentation
- L2 regularization
- Early stopping



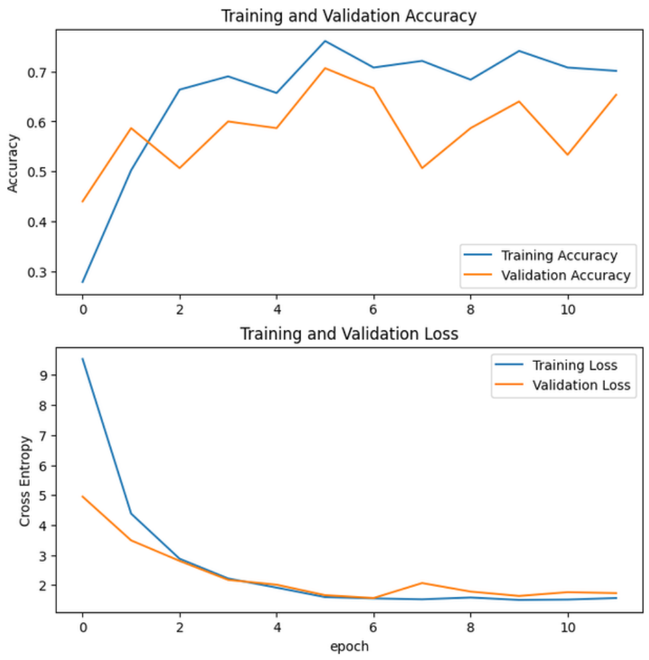
Data augmentation example

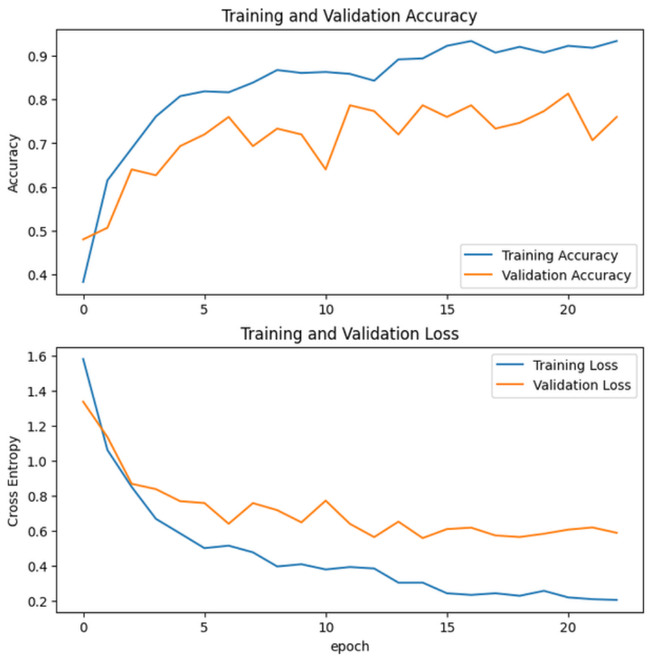
IMPLEMENTATION - PHILOVARIETY FINDER METHODOLOGY OF THE SYSTEM

4. Hyperparameter Tuning for the selected model

Before

Testing accuracy 76.32%





After

Testing accuracy 82.89%

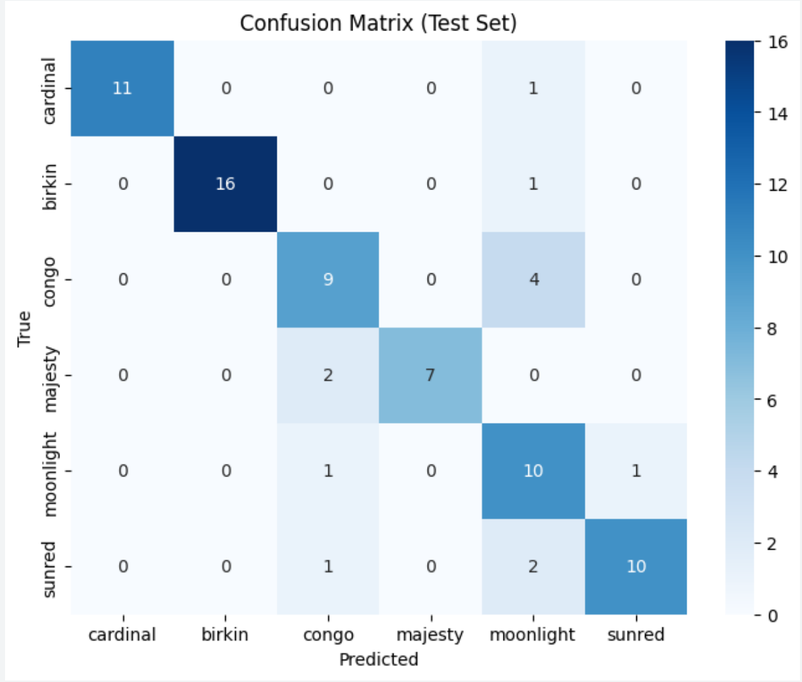
SLIIT FACULTY OF COMPUTING

IT19169736 | Gamage M.G.U.D. | 2023-133

IMPLEMENTATION - PHILOVARIETY FINDER METHODOLOGY OF THE SYSTEM

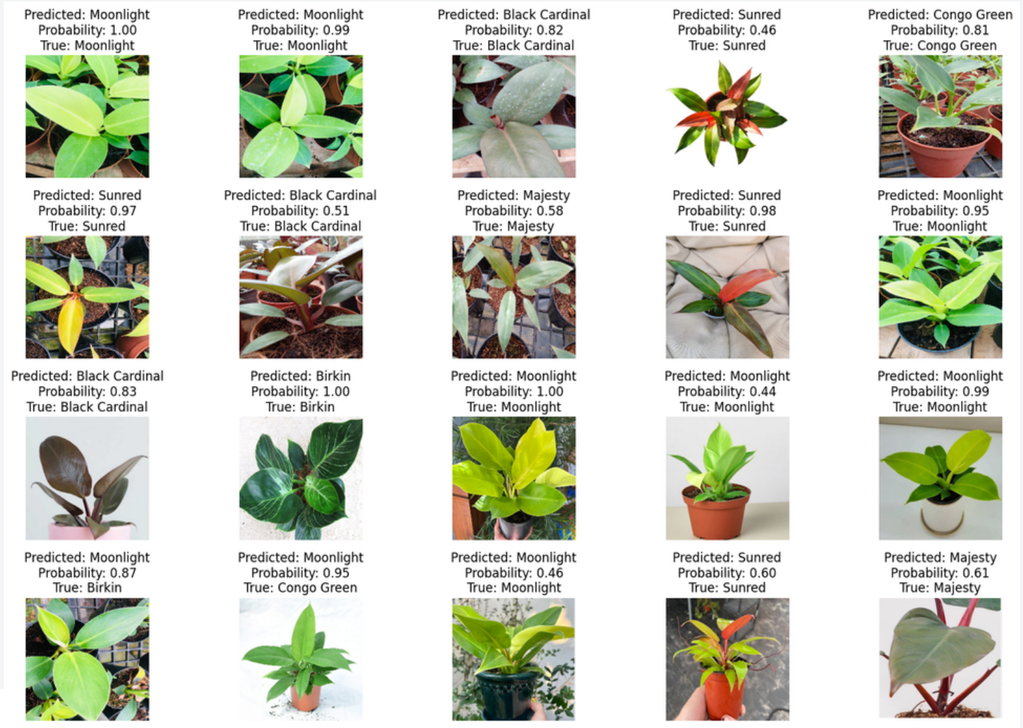
4. Model evaluation

Confusion Matrix



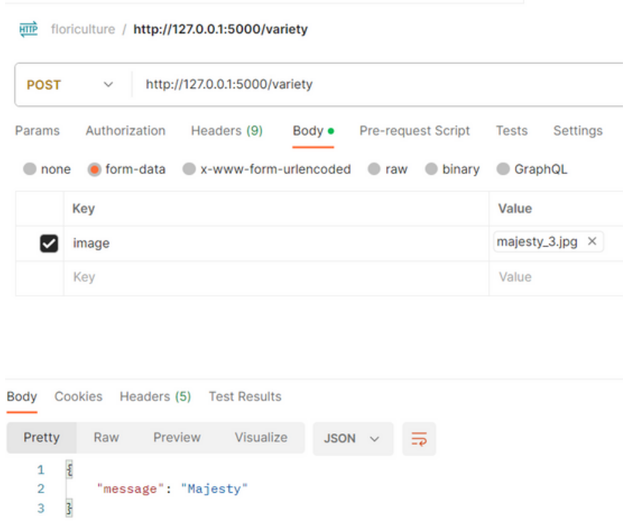
IMPLEMENTATION - PHILOVARIETY FINDER METHODOLOGY OF THE SYSTEM

5. Prediction on a test set

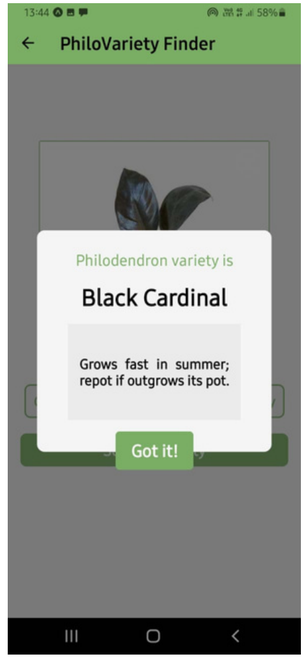


IMPLEMENTATION - PHILOVARIETY FINDER METHODOLOGY OF THE SYSTEM

6. Flask API & React Native mobile app



Testing Flask API on Postman



Output in React Native app

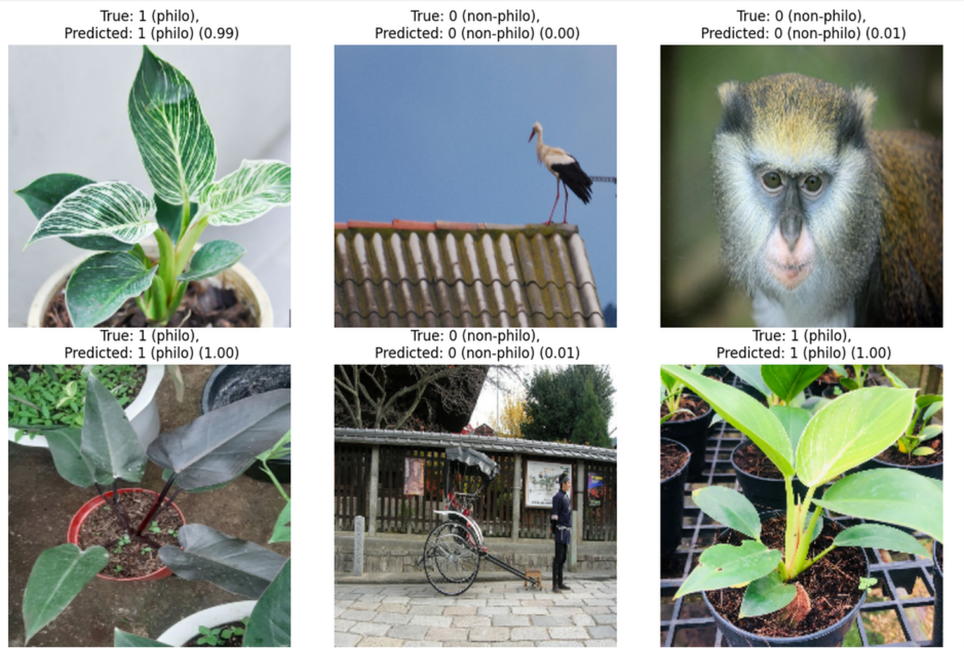
SLIIT FACULTY OF COMPUTING

IT19169736 | Gamage M.G.U.D. | 2023-133

IMPLEMENTATION - PHILOVARIETY FINDER METHODOLOGY OF THE SYSTEM

7. Binary Classifier for validation

- **MobileNetV2** pre-trained model is used as the feature extractor.
- And additional layers are added
- Classifies images into Philo / Non-Philo
- Enables validation in app

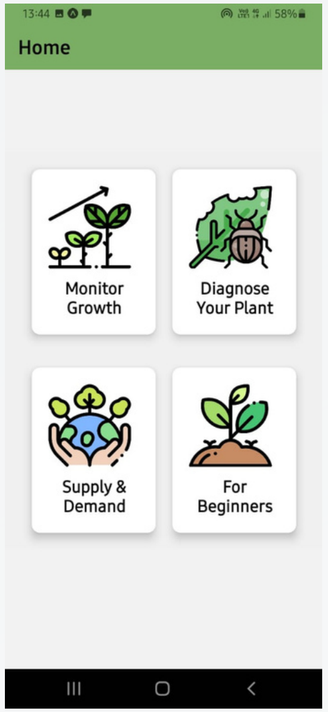


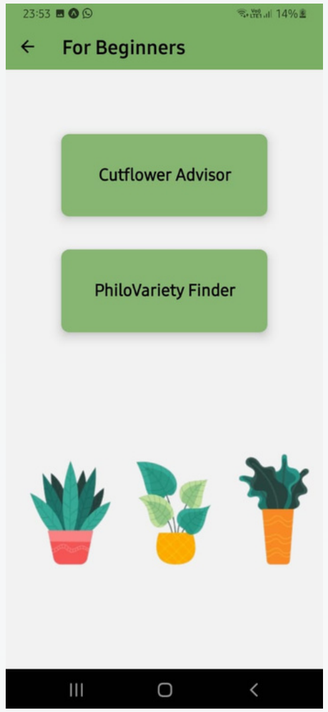
Prediction on a test set

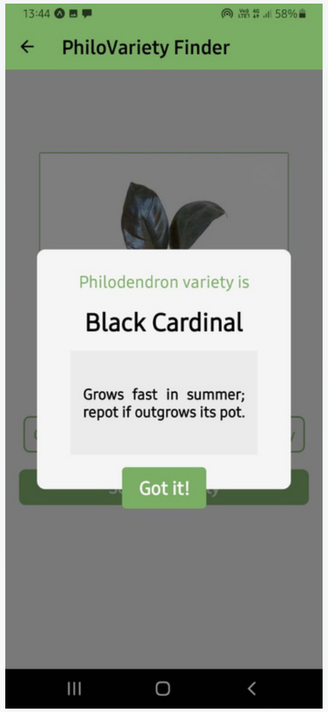
SLIIT FACULTY OF COMPUTING

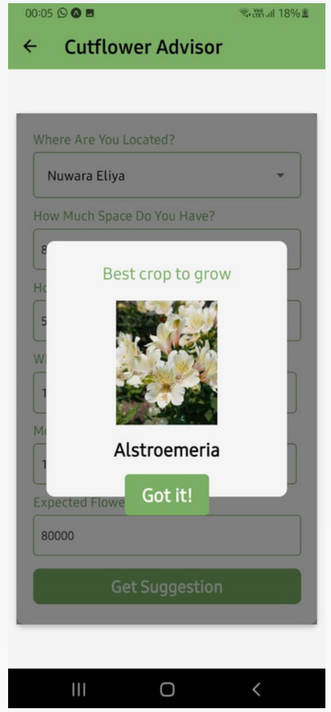
IT19169736 | Gamage M.G.U.D. | 2023-133

UI DESIGNS









SLIIT
FACULTY OF COMPUTING

IT19169736 | Gamage M.G.U.D. | 2023-133

NON-FUNCTIONAL REQUIREMENTS

- Reliable
- Fast
- Accurate
- User-friendly

FUTURE WORKS

Improve model accuracy

Test the app with the actual users

Adjust the functionalities
according to user requirements

SLIIT
FACULTY OF COMPUTING

IT19169736 | Gamage M.G.U.D. | 2023-133

IT20017088
Prabhashi P.A.N.

FORECAST THE FLUCTUATING DEMAND PATTERNS BASED ON SEASONS AND SPECIFIC PRODUCT PREFERENCES

SLIIT
FACULTY OF COMPUTING

Specialization : Data Science

RESEARCH PROBLEM

The floriculture industry depends on accurate demand forecasting for efficient production, inventory management, and customer satisfaction. Yet, demand prediction in this industry is intricate due to numerous dynamic factors. These factors include humidity, rainfall, temperature, country of sale, pot size, seasonality, and events, all impacting consumer preferences and buying habits.

SLIIT
FACULTY OF COMPUTING

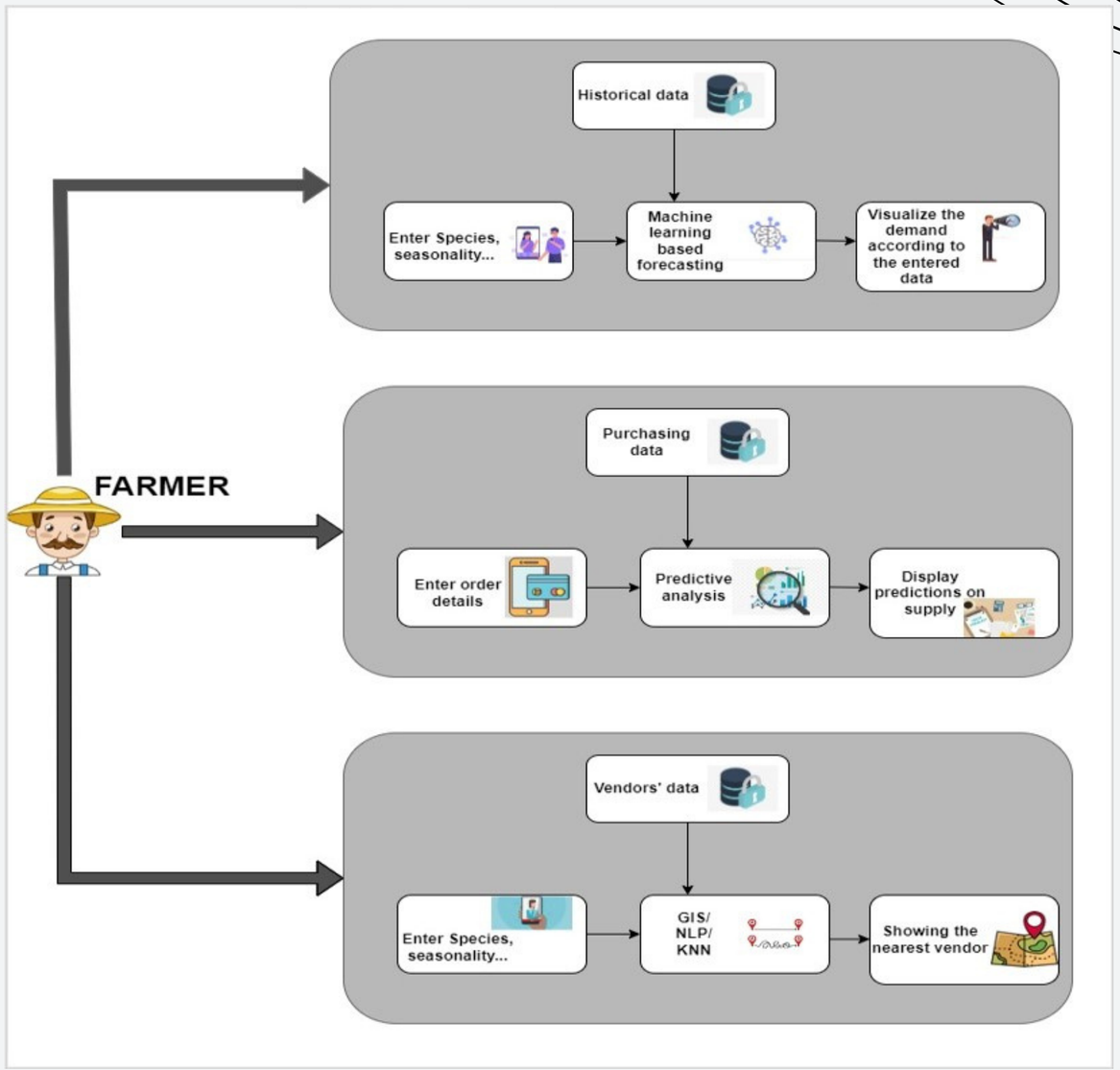
IT20017088 | Prabhashi P.A.N. | 2023-133

GAP BETWEEN OTHER STUDIES

- Demand forecasting for the market in agriculture sector crops
- Using LSTM or other time series analysis for demand forecasting

- Demand forecasting specially for the floriculture products with the special features
- Using better approaches like linear regression with the feature variety among the floriculture industry

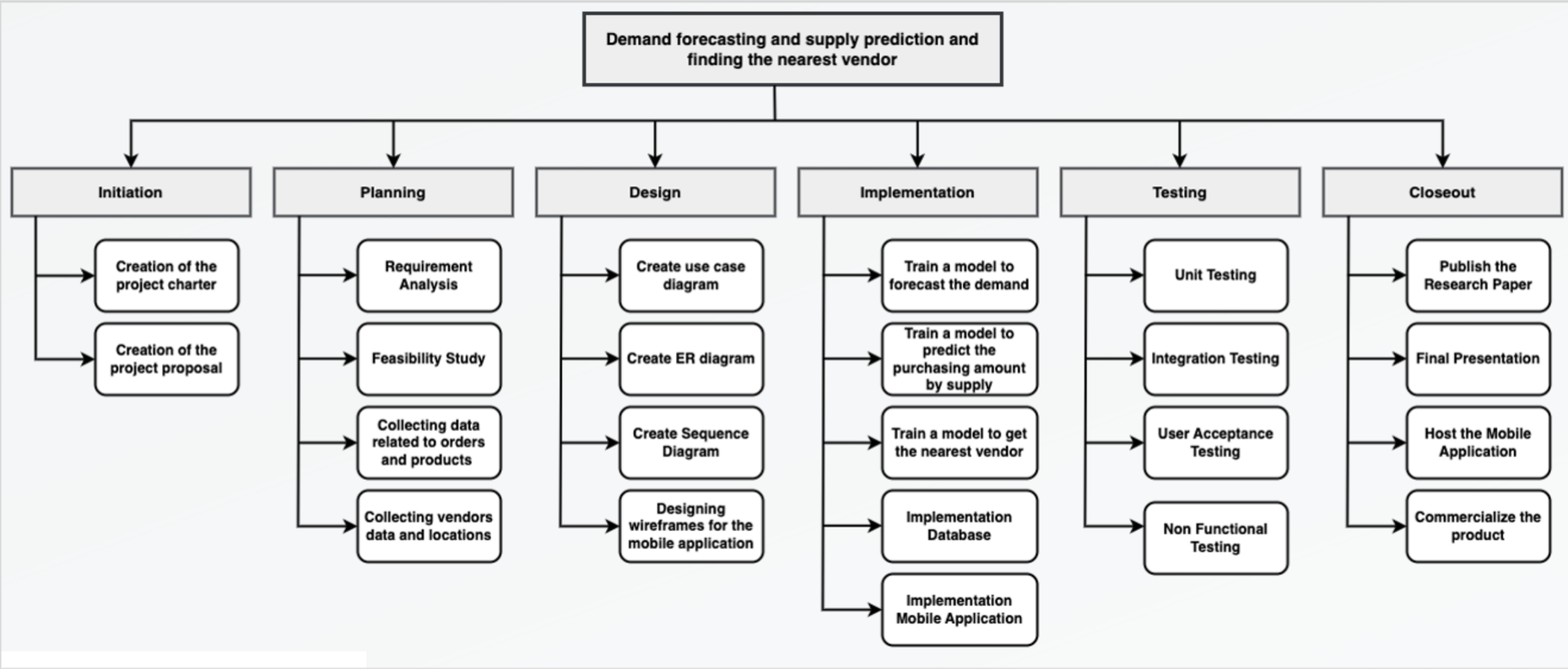
INDIVIDUAL SYSTEM DIAGRAM



SLIIT
FACULTY OF COMPUTING

IT20017088 | Prabhashi P.A.N. | 2023-133

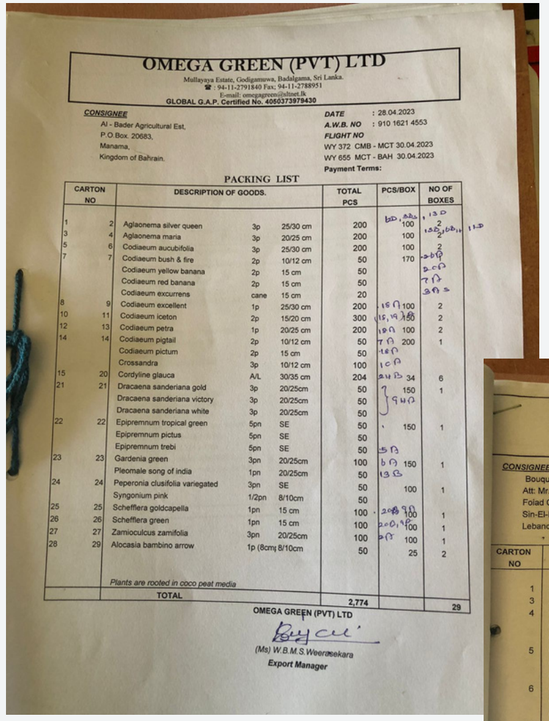
WORK BREAKDOWN



FIELD DATA

These data is collected from the omega green Pvt Ltd

And all data are collected in manual way



OMEGA GREEN (PVT) LTD

Mulleyaya Estate, Godigamuwa, Badalgama, Sri Lanka.
☎ : 94(1)-2791840 Fax: 94-11-2788951
E-mail: omegagreen@sltnet.lk
GLOBAL G.A.P. Certified No. 4060373979430

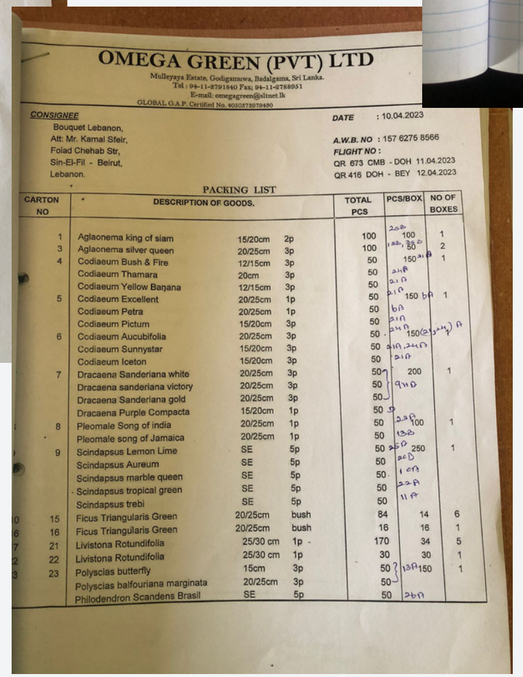
CONSIGNEE
Al - Bader Agricultural Est,
P.O Box. 20683,
Manama,
Kingdom of Bahrain.

DATE : 28.04.2023
A.W.B. NO : 910 1621 4553
FLIGHT NO
WY 372 CMB - MCT 30.04.2023
WY 655 MCT - BAH 30.04.2023
Payment Terms:

PACKING LIST

CARTON NO		DESCRIPTION OF GOODS.			TOTAL PCS	PCS/BOX	NO OF BOXES
1	2	Aglaonema silver queen	3p	25/30 cm	200	100	2
3	4	Aglaonema maria	3p	20/25 cm	200	100	2
5	6	Codiaeum aucubifolia	3p	25/30 cm	200	100	2
7	7	Codiaeum bush & fire	2p	10/12 cm	50	170	
		Codiaeum yellow banana	2p	15 cm	50		
		Codiaeum red banana	2p	15 cm	50		
		Codiaeum excurrens	cane	15 cm	20		
8	9	Codiaeum excellent	1p	25/30 cm	200	100	2
10	11	Codiaeum iceton	2p	15/20 cm	300	100	2
12	13	Codiaeum petra	1p	20/25 cm	200	100	2
14	14	Codiaeum pigtail	2p	10/12 cm	50	200	1
		Codiaeum pictum	2p	15 cm	50		
		Crossandra	3p	10/12 cm	100		
15	20	Cordyline glauca	A/L	30/35 cm	204	34	6
21	21	Dracaena sanderiana gold	3p	20/25cm	50	150	1
		Dracaena sanderiana victory	3p	20/25cm	50		
		Dracaena sanderiana white	3p	20/25cm	50		
22	22	Epipremnum tropical green	5pn	SE	50	150	1
		Epipremnum pictus	5pn	SE	50		
		Epipremnum trebi	5pn	SE	50		
23	23	Gardenia green	3pn	20/25cm	100	150	1
		Pleomale song of india	1pn	20/25cm	50		
24	24	Peperonia clusifolia variegated	3pn	SE	50	100	1
		Syngonium pink	1/2pn	8/10cm	50		
25	25	Scheffera goldcapella	1pn	15 cm	100	100	1
26	26	Scheffera green	1pn	15 cm	100	100	1
27	27	Zamioculcus zamifolia	3pn	20/25cm	100	100	1
28	29	Alocasia bambino arrow	1p (8cm)	8/10cm	50	25	2
		Plants are rooted in coco peat media					
		TOTAL			2,774		29

OMEGA GREEN (PVT) LTD
(Ms) W.B.M.S.Weerasekara
Export Manager



OMEGA GREEN (PVT) LTD

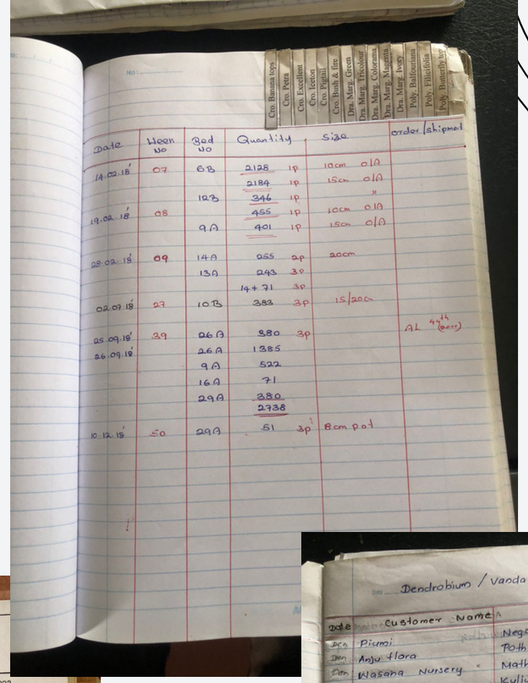
Mulleyaya Estate, Godigamuwa, Badalgama, Sri Lanka.
Tel : 94-11-2791840 Fax: 94-11-2788951
E-mail: omegagreen@sltnet.lk
GLOBAL G.A.P. Certified No. 4060373979430

CONSIGNEE
Bouquet Lebanon,
Att: Mr. Kamal Sfeir,
Folad Chehab Str,
Sin-El-Fil - Beirut,
Lebanon.

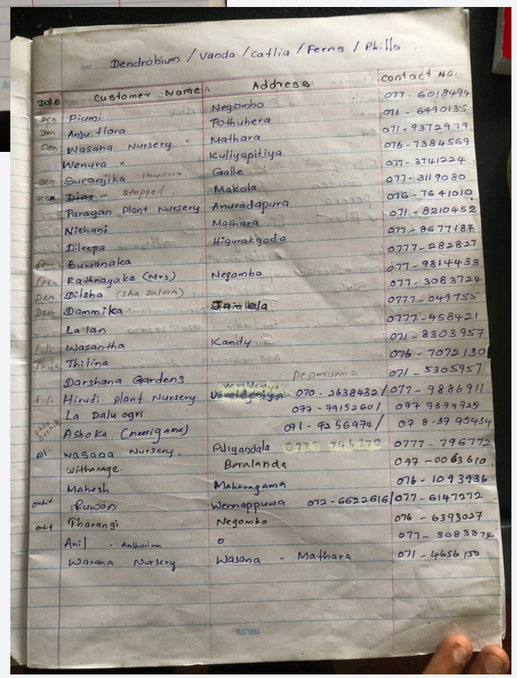
DATE : 10.04.2023
A.W.B. NO : 157 6275 8566
FLIGHT NO :
QR 673 CMB - DOH 11.04.2023
QR 416 DOH - BEY 12.04.2023

PACKING LIST

CARTON NO		DESCRIPTION OF GOODS.			TOTAL PCS	PCS/BOX	NO OF BOXES
1		Aglaonema king of siam	15/20cm	2p	100	100	1
3		Aglaonema silver queen	20/25cm	3p	100	100	2
4		Codiaeum Bush & Fire	12/15cm	3p	50	150	1
		Codiaeum Thamara	20cm	3p	50		
		Codiaeum Yellow Banana	12/15cm	3p	50		
5		Codiaeum Excellent	20/25cm	1p	50	150	1
		Codiaeum Petra	20/25cm	1p	50		
		Codiaeum Pictum	15/20cm	3p	50		
6		Codiaeum Aucubifolia	20/25cm	3p	50	150	1
		Codiaeum Sunnystar	15/20cm	3p	50		
		Codiaeum Iceton	15/20cm	3p	50		
7		Dracaena Sanderiana white	20/25cm	3p	50	200	1
		Dracaena sanderiana victory	20/25cm	3p	50		
		Dracaena Sanderiana gold	20/25cm	3p	50		
		Dracaena Purple Compacta	15/20cm	1p	50		
8		Pleomale Song of india	20/25cm	1p	50	100	1
		Pleomale song of Jamaica	20/25cm	1p	50		
9		Scindapsus Lemon Lime	SE	5p	50	250	1
		Scindapsus Aureum	SE	5p	50		
		Scindapsus marble queen	SE	5p	50		
		Scindapsus tropical green	SE	5p	50		
		Scindapsus trebi	SE	5p	50		
15		Ficus Triangularis Green	20/25cm	bush	84	14	6
16		Ficus Triangularis Green	20/25cm	bush	16	16	1
21		Livistona Rotundifolia	25/30 cm	1p	170	34	5
22		Livistona Rotundifolia	25/30 cm	1p	30	30	1
23		Polyscias butterfly	15cm	3p	50	150	1
		Polyscias balfouriana marginata	20/25cm	3p	50		
		Philodendron Scandens Brasil	SE	5p	50		



Date	Item No	Bed No	Quantity	Size	order/shipment
14.02.18	07	6B	2128 1p	10cm o/A	
			2184 1p	15cm o/A	
19.02.18	08	12B	346 1p	"	
		9A	455 1p	10cm o/A	
			401 1p	15cm o/A	
20.02.18	09	14A	265 3p	20cm	
		13A	293 3p		
			147 3p		
02.07.18	27	10B	383 3p	15/20cm	
25.09.18	29	26A	880 3p		AL 4th (berry)
26.09.18		26A	1385		
		9A	522		
		16A	71		
		29A	380		
			2938		
10.12.18	50	29A	51 3p	8cm pot	



Dendrobium / Vanda / cattla / Ferns / Philo

Date	Customer Name	Address	Contact No:
Jan	Pismi	Negombo	077 - 6018494
Jan	Anju Flora	Polhubera	071 - 6491035
Jan	Wasana Nursery	Mathara	071 - 9372977
Jan	Wenura "	Kuliyapitiya	076 - 7384569
Jan	Suranjika Pandiram	Galle	077 - 3741224
Jan	Dias - stopped	Makola	077 - 3119080
Jan	Paragan Plant Nursery	Anuradapura	076 - 7641010
Jan	Nishani	Mathara	071 - 8210452
Jan	Dileepa	Higurakgoda	077 - 9677187
Jan	Bussanaka		0777 - 282827
Feb	Rathnayake (Mrs)	Negombo	077 - 9814438
Feb	Dilsha (Isha Salon)		077 - 3083724
Feb	Dammika	Jaela	0777 - 049755
Feb	Lalan		0777 - 458421
Feb	Wasantha	Kandy	071 - 8303957
Feb	Thilina		076 - 7072130
Feb	Darshana Gardens	Pepiliyana	071 - 5305957
Feb	Hiruti plant Nursery	Dewalegama	070 - 3638432 / 077 - 9886911
Feb	La Dalu agri		072 - 7915260 / 077 9899739
Feb	Ashoka (Meerigama)		071 - 9256974 / 07 8-39 95454
Mar	Wasana Nursery	Piliyandala	0777 - 796772
Mar	Withange	Boralande	077 - 0063610
Mar	Mahesh	Maharagama	076 - 1093986
Mar	Sluvan	Wennappuwa	072 - 6622616 / 077 - 6147272
Mar	Dharangi	Negombo	076 - 6393027
Mar	Anil - Anthurium	o	077 - 9083878
Mar	Wasana Nursery	Wasana - Mathara	071 - 4656150

Using all these manually arranged data I have created newest version of data set as required

SLIIT FACULTY OF COMPUTING

IT20017088 | Prabhashi P.A.N. | 2023-133

IMPLEMENTATION

Data Preprocessing

Checking for the null values and data types

Converting related data in to the needed types

```
1 data
```

	DateOfSale	Temperature	Humidity	Rainfall	Philo Sando
0	1/1/2012	27.5°C	81	227.57	8950
1	1/2/2012	25.7°C	80	166.40	6930
2	1/3/2012	26.3°C	78	147.26	2470
3	1/4/2012	25.2°C	81	191.72	2420

```
1 # Handle missing values
2 # Check for missing values in each column
3 data.isnull().sum() #no missing values so far
```

```
DateOfSale          0
Temperature         0
Humidity            0
Rainfall            0
Philo Sando         0
Aglonema            0
Lemon lime          0
Dendrobium          0
Ferns               0
Zamioculus          0
Bengamina           0
Dracaena            0
Country             0
PotSize             0
Season              0
Event               0
HighestSalesPlant   0
dtype: int64
```

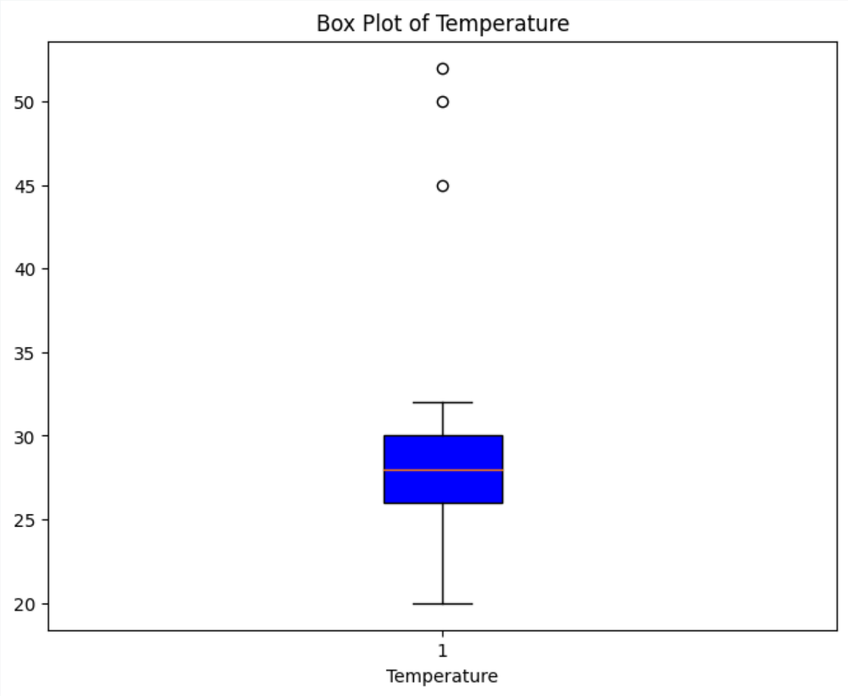
+ Code + Text

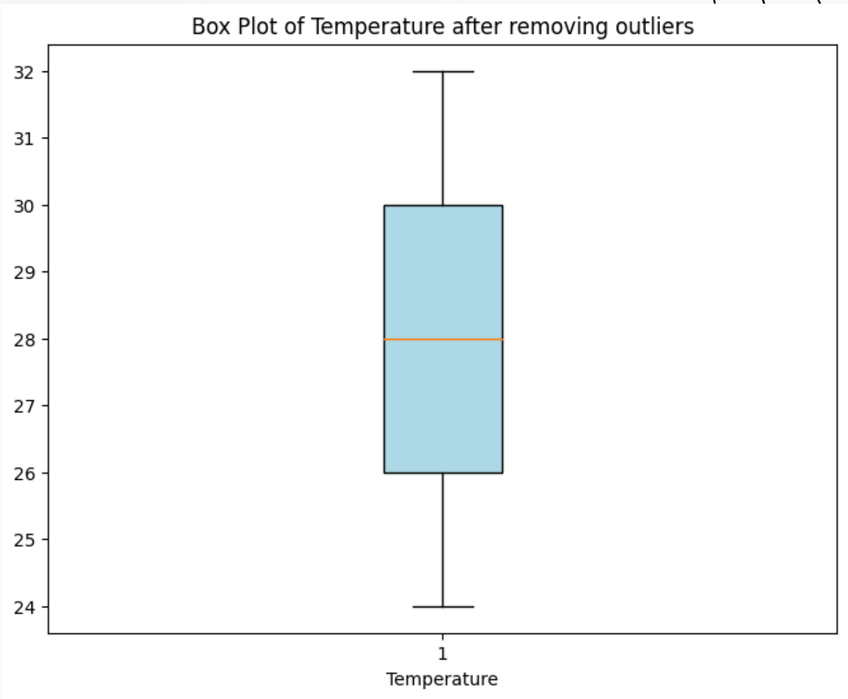
```
[14]  1 # Convert "DateOfSale" column to datetime type
      2 data['DateOfSale'] = pd.to_datetime(data['DateOfSale'])
      3 # Extract numerical part from Temperature and convert to numeric type
      4 data['Temperature'] = data['Temperature'].str.extract('(\d+)').astype(float)
```

IMPLEMENTATION

Data Visualizing for the outliers checkers

Temperature

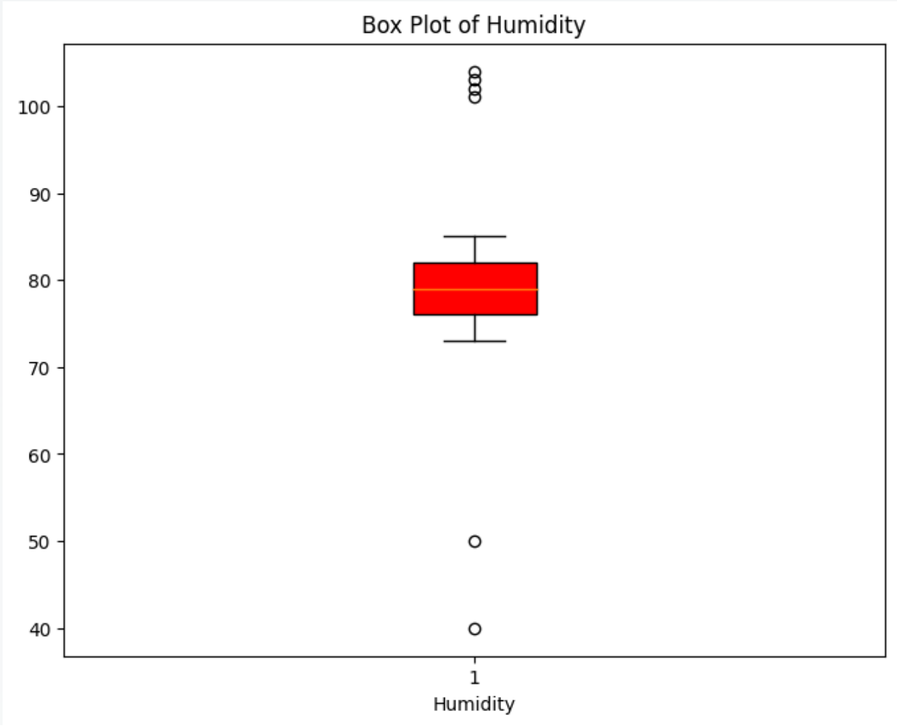


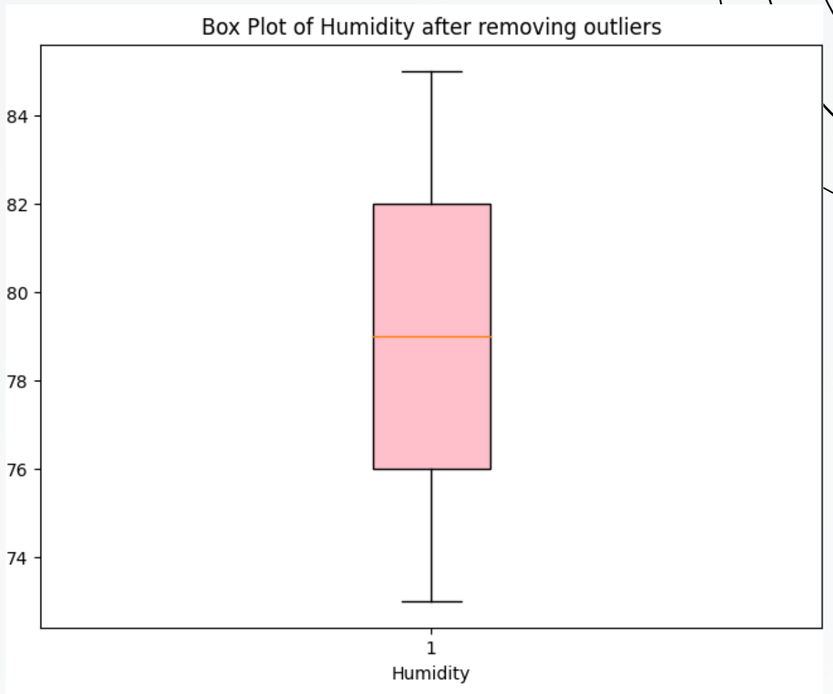


IMPLEMENTATION

Data Visualizing for the outliers checkers

Humidity

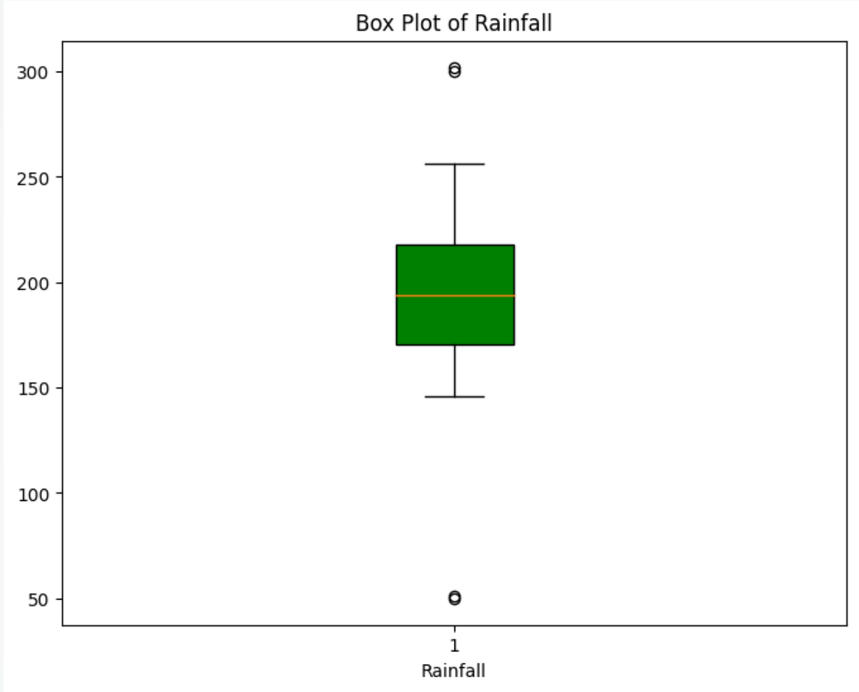


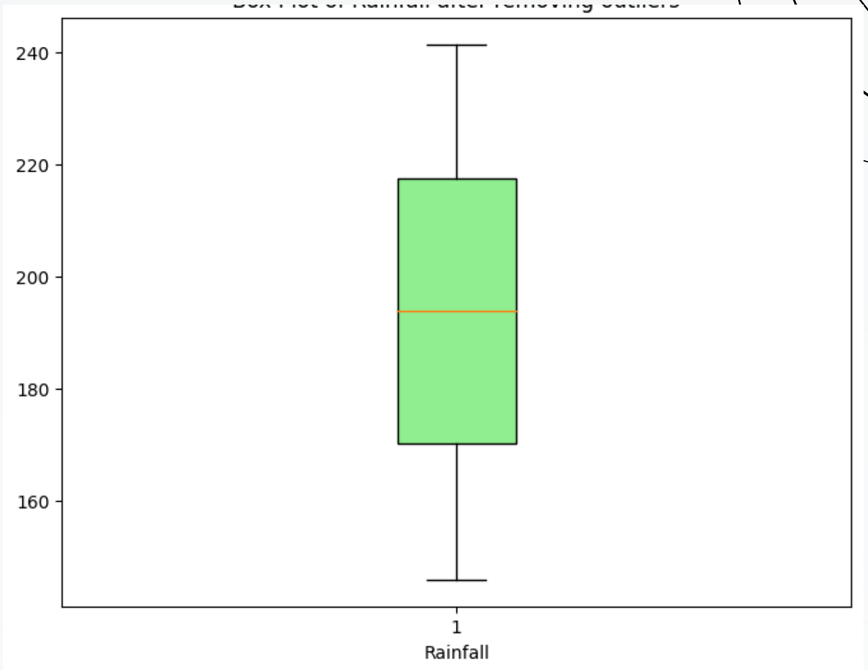


IMPLEMENTATION

Data Visualizing for the outliers checkers

Rainfall





IMPLEMENTATION

Outliers checking and stabalize

Applying square root transformation for stabalize the variance

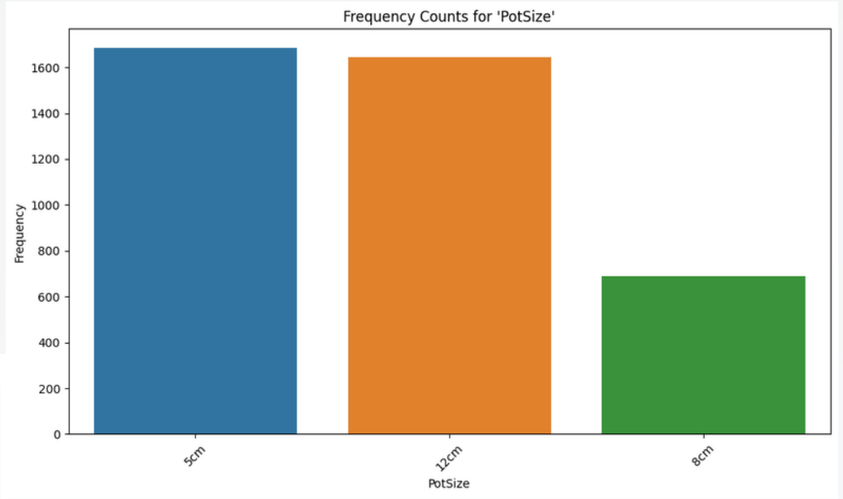
```
1 for column in plants_columns_data:
2     # Applying square root transformation
3     transformed_data[f"sqrt_{column}"] = np.sqrt(plants_columns_data[column])
4
```

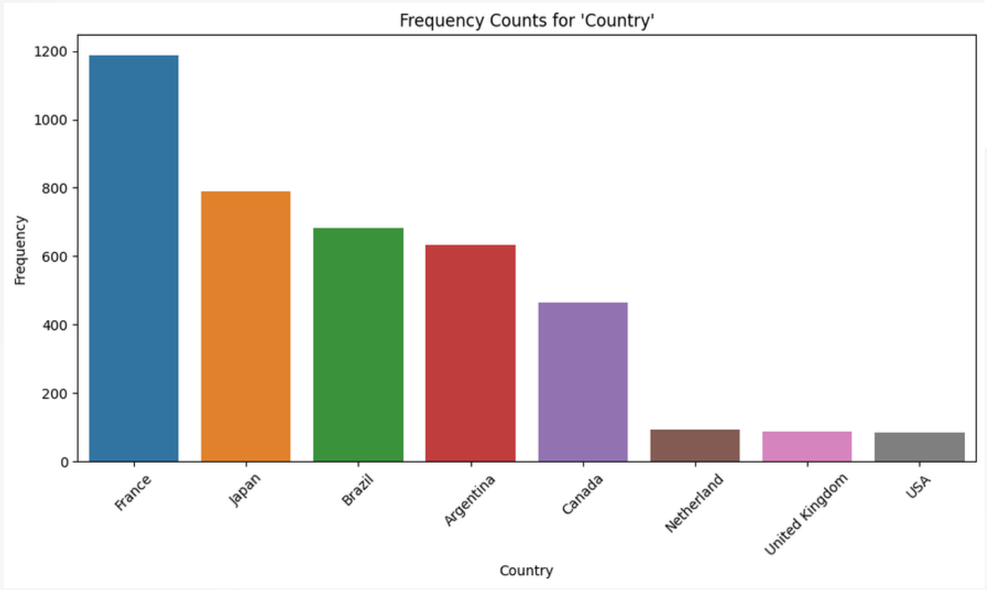
```
1 transformed_data
```

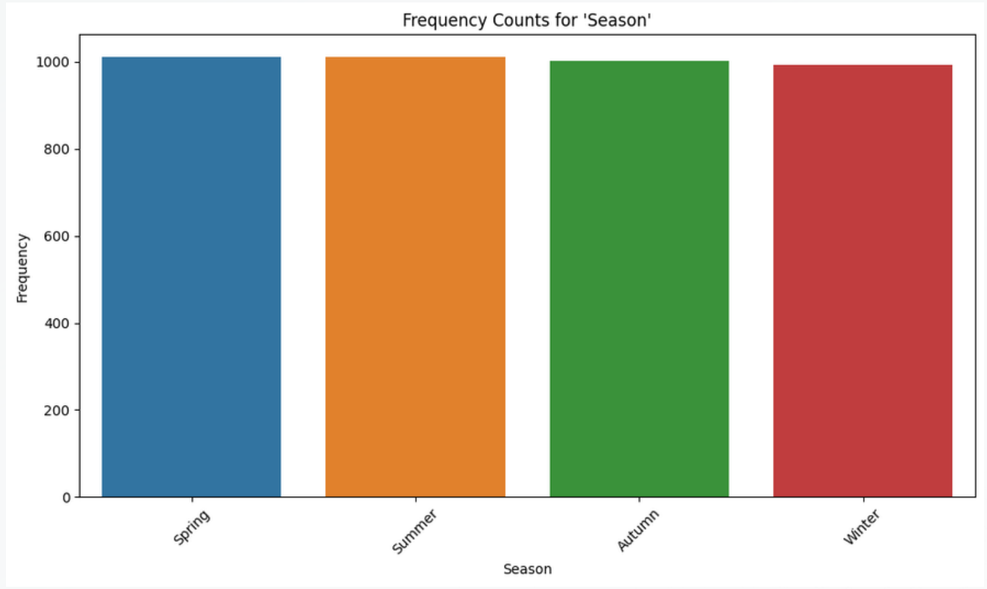
	sqrt_Philo Sando	sqrt_Aglonema	sqrt_Lemon lime	sqrt_Dendrobium	sqrt_Ferns	sqrt_Zamioculus	sqrt_Bengamina	sqrt_Draca
0	94.604440	49.091751	62.289646	60.166436	50.099900	47.749346	53.851648	60.745
1	83.246622	63.245553	55.767374	48.579831	57.183914	57.183914	54.497706	53.944
2	49.699095	58.137767	92.574294	59.413803	51.478151	57.358522	50.299105	57.358
3	49.193496	49.396356	94.233752	52.535702	59.749477	60.249481	51.672043	48.062
4	57.183914	77.395090	56.213877	52.440442	48.989795	50.695167	55.497748	60.332

IMPLEMENTATION

Data Visualizing for frequency checking





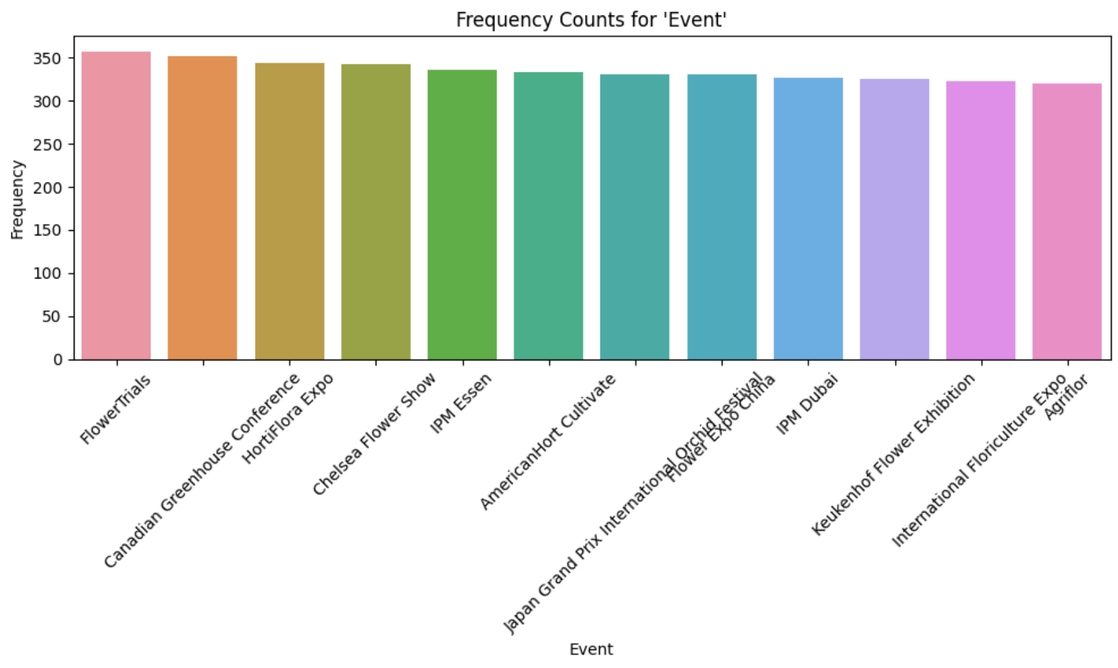


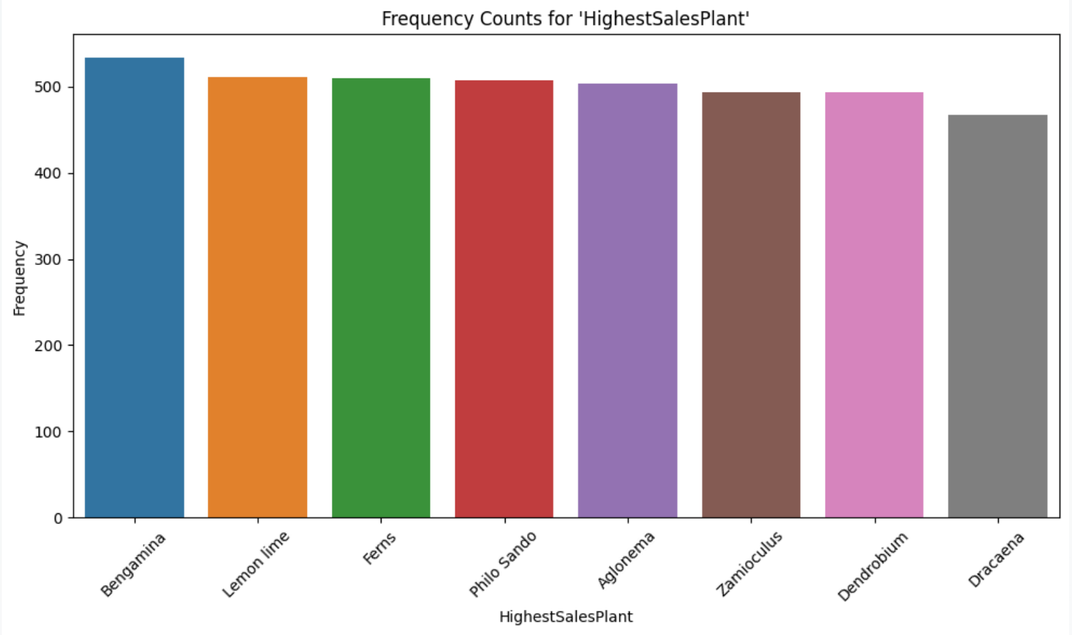
SLIIT FACULTY OF COMPUTING

IT20017088 | Prabhashi P.A.N. | 2023-133

IMPLEMENTATION

Data Visualizing for frequency checking





SLIIT
FACULTY OF COMPUTING

IT20017088 | Prabhashi P.A.N. | 2023-133

IMPLEMENTATION

EDA(Exploratory Data Analysis

Exploratory Data Analysis (EDA)

```
[56]  1 data_withoutOutliers
```

	DateOfSale	Temperature	Humidity	Rainfall	Country	PotSize	Season	Event
0	1/1/2012	27.0	81.0	227.57	France	5cm	Winter	IPM Dubai
1	1/2/2012	25.0	80.0	166.40	France	5cm	Winter	Japan Grand Prix International Orchid Festival
2	1/3/2012	26.0	78.0	147.26	France	8cm	Winter	IPM Dubai
3	1/4/2012	25.0	81.0	191.72	United Kingdom	5cm	Winter	IPM Essen
4	1/5/2012	27.0	77.0	236.65	Argentina	12cm	Winter	Japan Grand Prix International Orchid Festival

```
1 # Display summary statistics
2 summary_stats = data_withoutOutliers.describe()
3 print(summary_stats)
```

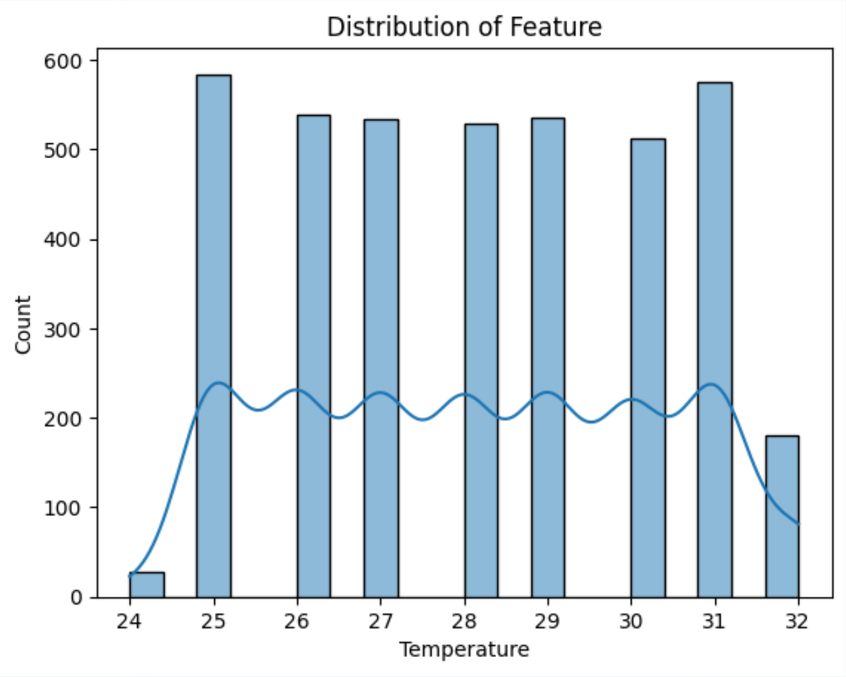
```
       Temperature     Humidity     Rainfall  sqrt_Philo Sando  sqrt_Aglonema  \
count  4018.000000  4018.000000  4018.000000       4018.000000    4018.000000
mean     28.132550    79.031415   193.991437         58.175267      58.051612
std       2.173201     3.742590    27.482668         10.987635      11.192096
min      24.000000    73.000000   146.070000         44.721360      44.721360
25%      26.000000    76.000000   170.395000         50.695167      50.497525
50%      28.000000    79.000000   193.991318         56.035703      55.767374
75%      30.000000    82.000000   217.502500         60.971294      60.991803
max      32.000000    85.000000   241.430000         94.868330      94.868330

       sqrt_Lemon lime  sqrt_Dendrobium   sqrt_Ferns  sqrt_Zamioculus  \
count      4018.000000      4018.000000  4018.000000      4018.000000
mean         58.125710        58.041783    58.203475        58.054577
std          11.029104        11.056090    11.081927        11.131619
min          44.721360        44.721360    44.721360        44.721360
25%          50.793700        50.596443    50.793700        50.497525
50%          55.946403        56.124861    56.213877        56.035703
75%          60.991803        60.827625    60.991803        60.991803
max          94.868330        94.815611    94.868330        94.815611

       sqrt_Bengamina  sqrt_Dracaena
count     4018.000000    4018.000000
mean        58.347053      57.817594
std         11.240872      10.701687
```

SLIIT FACULTY OF COMPUTING

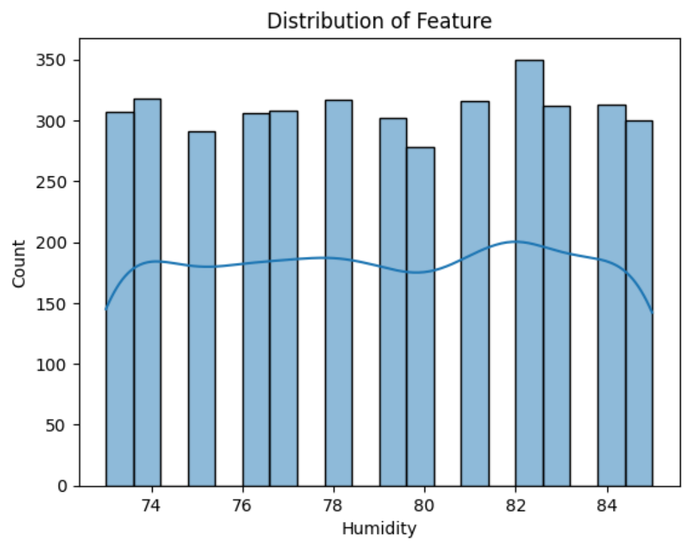
IT20017088 | Prabhashi P.A.N. | 2023-133

IMPLEMENTATION

EDA(Exploratory Data Analysis

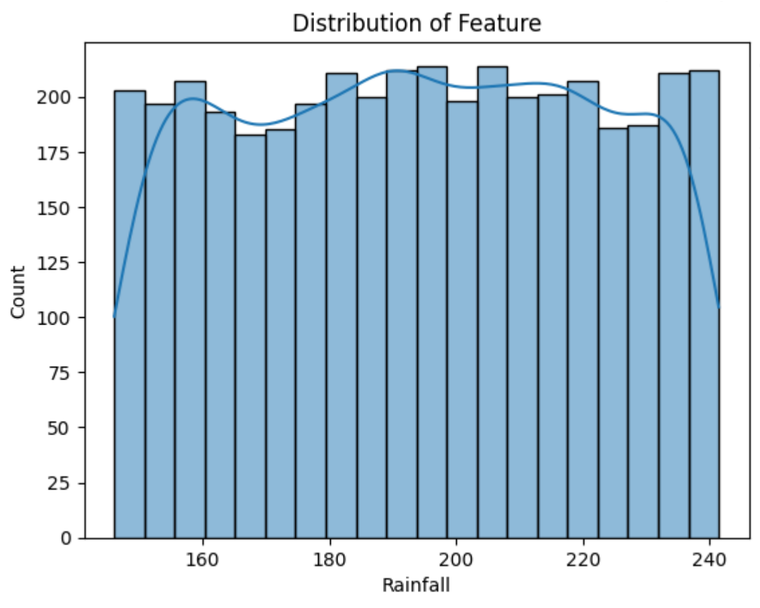


Temperature



Humidity

Distributions



Rainfall

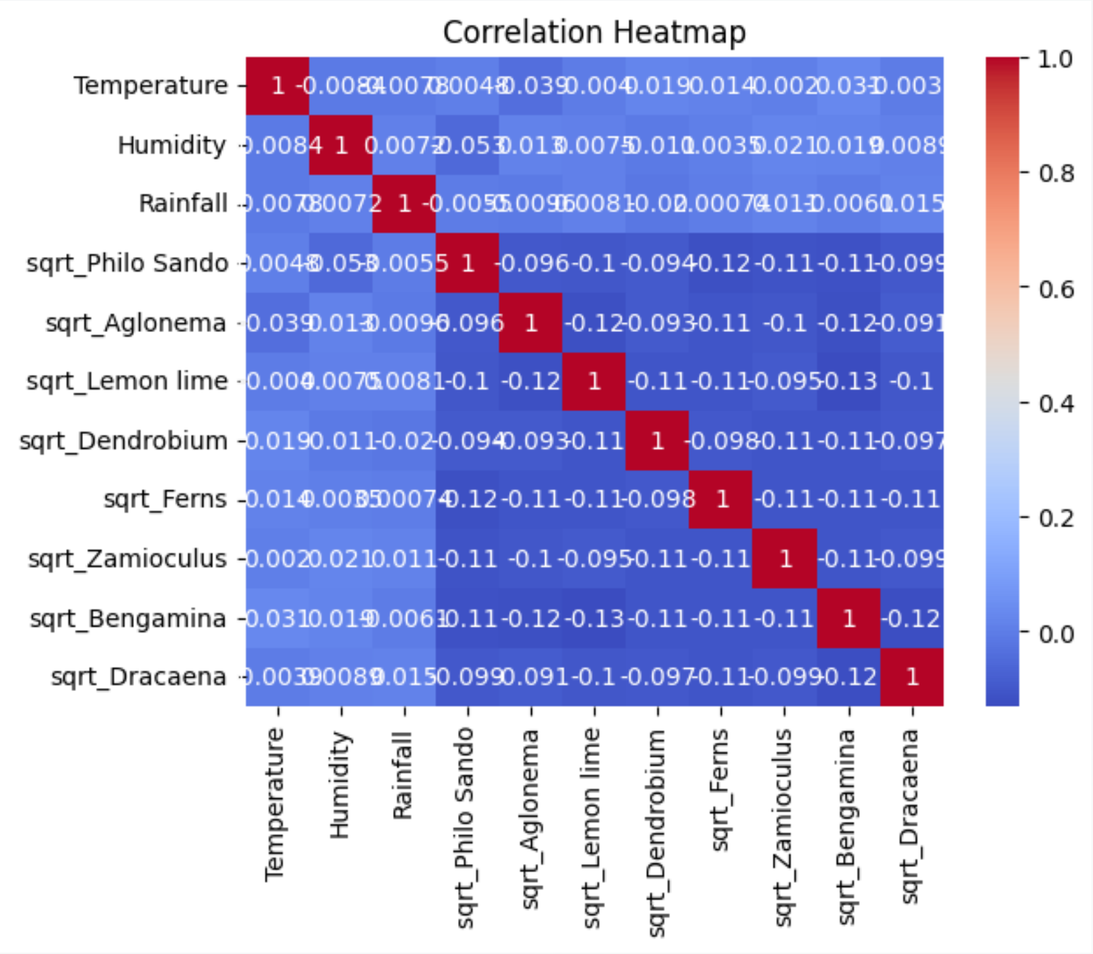
SLIIT FACULTY OF COMPUTING

IT20017088 | Prabhashi P.A.N. | 2023-133

IMPLEMENTATION

EDA(Exploratory Data Analysis

Correlations



SLIIT FACULTY OF COMPUTING

IMPLEMENTATION

Data Preprocessing

Numerical data scaled using the minmax scaler and the categorical variables handled by one hot encoding

```
1 # Fit and transform using the min-max scaler
2 scaled_numerical_data_minmax = minmax_scaler.fit_transform(numerical_model_data[numerical_model_columns])
```

```
[74] 1
     2 scaled_numerical_data_minmax_df = pd.DataFrame(scaled_numerical_data_minmax, columns=numerical_model_columns)
```

```
[75] 1 scaled_numerical_data_minmax_df
```

	Temperature	Humidity	Rainfall	sqrt_Philo Sando	sqrt_Aglonema	sqrt_Lemon lime	sqrt_Dendrobium	sqrt_Ferns	sqrt_Zamioculus
0	0.375	0.666667	0.854656	0.994738	0.087152	0.350336	0.308320	0.107256	0.060446
1	0.125	0.583333	0.213192	0.76824					
2	0.250	0.416667	0.012479	0.09926					
3	0.125	0.666667	0.478712	0.08918					
4	0.375	0.333333	0.949874	0.24852					
...	...	...	...						
4013	0.625	0.833333	0.856334	0.65284					

```
1 onehot_encoded_categorical_data
```

	Country_Argentina	Country_Brazil	Country_Canada	Country_France	Country_Japan	Country_Netherland	C
0	0	0	0	1	0	0	
1	0	0	0	1	0	0	
2	0	0	0	1	0	0	
3	0	0	0	0	0	0	
4	1	0	0	0	0	0	
...	...	...	...	...	...	...	
4013	0	0	0	1	0	0	
4014	1	0	0	0	0	0	
4015	1	0	0	0	0	0	
4016	1	0	0	0	0	0	
4017	0	0	0	0	1	0	

IMPLEMENTATION

Splitting data in to x & y

The data set has split in to two as independent and dependent variables by 30% of the data taken as the test size

```
# Define the target columns and feature columns
target_columns = ['sqrt_Philo Sando', 'sqrt_Aglonema', 'sqrt_Lemon lime', 'sqrt_Dendrobium', 'sqrt_Ferns', 'sq
feature_columns = ['PotSize_12cm', 'PotSize_5cm', 'PotSize_8cm','Temperature', 'Humidity', 'Rainfall']

#feature_columns = ['Country_Argentina', 'Country_Brazil', 'Country_Canada', 'Country_France', 'Country_Japan'
```

```
[111]
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
import numpy as np

# Split the data into training and testing sets
X_train_all = final_model_data[feature_columns]  # Assuming your DataFrame is named model_final_data
y_train_all = final_model_data[target_columns]

# Assuming you want to split your data into a 70-30 ratio for training and testing
X_train, X_test, y_train, y_test = train_test_split(X_train_all, y_train_all, test_size=0.3, random_state=42)
```

SLIIT
FACULTY OF COMPUTING

IMPLEMENTATION

As for the progress presentation-1 I have implemented Random forest regression / Linear Regression / XGBoost regression and according to the accuracy I have taken the XGBoost regression as final

MODEL TRAINING

01.Random forest regression

```
[67]  1 # Random Forest Regression
      2 rf = RandomForestRegressor()
      3 rf.fit(X_train, y_train)
      4 rf_pred_train = rf.predict(X_train)
      5 rf_pred_test = rf.predict(X_test)
      6 rf_mse_train = mean_squared_error(y_train, rf_pred_train)
      7 rf_mse_test = mean_squared_error(y_test, rf_pred_test)
      8 print("Random Forest - Train MSE:", rf_mse_train)
      9 print("Random Forest - Test MSE:", rf_mse_test)

Random Forest - Train MSE: 0.9526527655989198
Random Forest - Test MSE: 1.13248521470431
```

03.Lineare Regression

```
[73]  1 # Linear Regression
      2 lr = LinearRegression()
      3 lr.fit(X_train, y_train)
      4 lr_pred_train = lr.predict(X_train)
      5 lr_pred_test = lr.predict(X_test)
      6 lr_mse_train = mean_squared_error(y_train, lr_pred_trai
      7 lr_mse_test = mean_squared_error(y_test, lr_pred_test)
      8 print("Linear Regression - Train MSE:", lr_mse_train)
      9 print("Linear Regression - Test MSE:", lr_mse_test)

Linear Regression - Train MSE: 0.9544264394774482
Linear Regression - Test MSE: 1.1360368381360995
```

02.XGBoost Regression

```
[70]  1 # XGBoost Regression
      2 xgb = XGBRegressor()
      3 xgb.fit(X_train, y_train)
      4 xgb_pred_train = xgb.predict(X_train)
      5 xgb_pred_test = xgb.predict(X_test)
      6 xgb_mse_train = mean_squared_error(y_train, xg
      7 xgb_mse_test = mean_squared_error(y_test, xgb_
      8 print("XGBoost - Train MSE:", xgb_mse_train)
      9 print("XGBoost - Test MSE:", xgb_mse_test)

XGBoost - Train MSE: 0.9526493973594272
XGBoost - Test MSE: 1.1325992861367322
```

IMPLEMENTATION

But as for the Progress Presentation-2 I have selected various techniques to find the ideal model for this

1.Random Forest Regressor

2.Gradient Boosting Regressor

3.Linear Regression

4.MLPRegressor

5.SARIMA as Time series forecasting

IMPLEMENTATION

Why these models:

1.Random Forest Regressor - Capture relations with data/ handle both categorical and numerical
2.Gradient Boosting Regressor-Excels predictive accuracy
3.Linear Regression
4.MLPRegressor- Can adapt to various patterns
5.SARIMA as Time series forecasting - Capture seasonal specific data

IMPLEMENTATION

Finally I have chosen Linear Regression and Why?

Provide the highest accuracy rather than other models
Handle both categorical and numerical values respectively
Robust to outliers
Effectively utilize historical data
Multivariate linear regression allows to simultaneously predict multiple targets based on the same set of predictor variables

IMPLEMENTATION

Linear Regression

```python
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import numpy as np

# Split the data into training and testing sets
X_train_all = final_model_data[feature_columns]  # Assuming your DataFrame
y_train_all = final_model_data[target_columns]

# Assuming you want to split your data into a 70-30 ratio for training and
X_train, X_test, y_train, y_test = train_test_split(X_train_all, y_train_a

# Loop through each target column and train Linear Regression
for target_column in target_columns:
    y_train_single = y_train[target_column]  # Extract the target column f
    y_test_single = y_test[target_column]    # Extract the target column f

    # Initialize the Linear Regression model
    linear_regression_model = LinearRegression()

    # Train the model
    linear_regression_model.fit(X_train, y_train_single)

    # Predict on the test set
    y_pred = linear_regression_model.predict(X_test)

    # Calculate RMSE
    rmse = np.sqrt(mean_squared_error(y_test_single, y_pred))
    print(f'Linear Regression RMSE for {target_column}: {rmse}')
```

```
Linear Regression RMSE for sqrt_Philo Sando: 0.22013772337975313
Linear Regression RMSE for sqrt_Aglonema: 0.2291142837625918
Linear Regression RMSE for sqrt_Lemon lime: 0.22067160672996564
Linear Regression RMSE for sqrt_Dendrobium: 0.21463910608934744
Linear Regression RMSE for sqrt_Ferns: 0.2248599049664761
Linear Regression RMSE for sqrt_Zamioculus: 0.21576147872871673
Linear Regression RMSE for sqrt_Bengamina: 0.21809673657828327
Linear Regression RMSE for sqrt_Dracaena: 0.21842194057587186
```

IMPLEMENTATION

After model evaluation part according to MAE, MSE and R squared values Linear regressor is the best model for this

Using accuracy calculated for all models and then choose the best

```
Best models for each target column:
sqrt_Philo Sando     Gradient Boosting
sqrt_Aglonema        Linear Regression
sqrt_Lemon lime      Linear Regression
sqrt_Dendrobium      Linear Regression
sqrt_Ferns           Linear Regression
sqrt_Zamioculus      Linear Regression
sqrt_Bengamina          Neural Network
sqrt_Dracaena        Linear Regression
dtype: object
```

```
[ ]  1 # the best approach is Linear regression
```

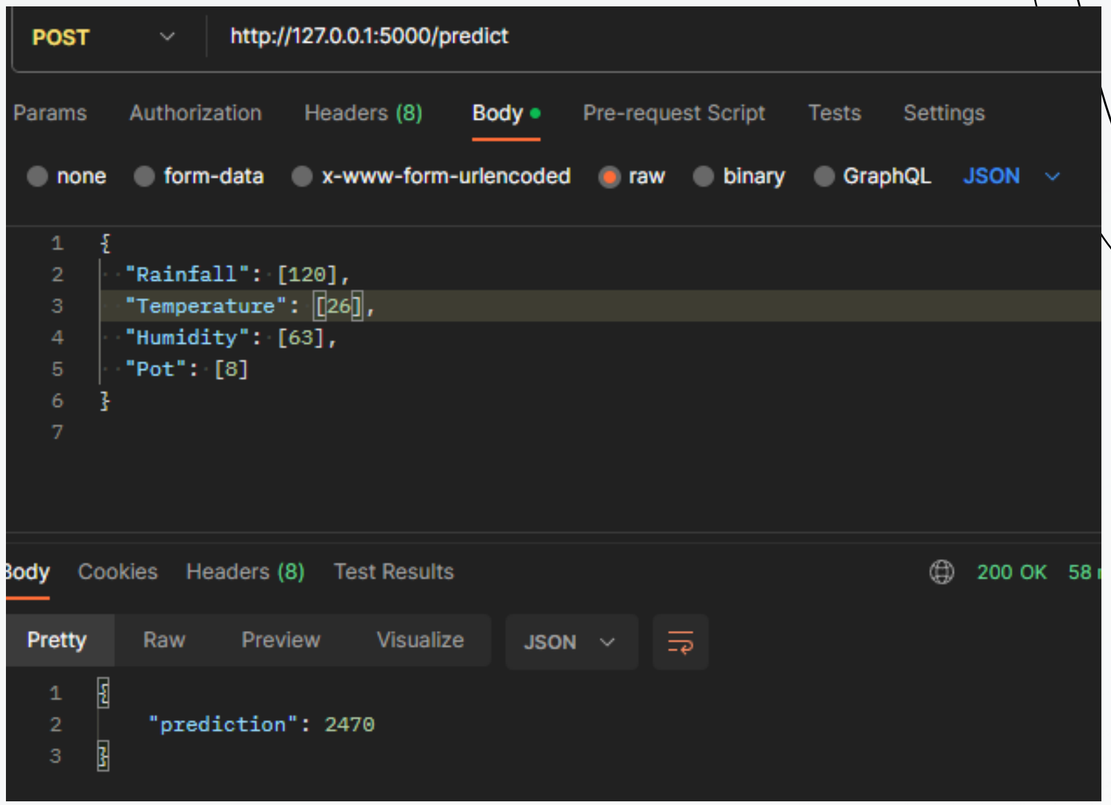
After figure out the model created a respective flask API and React native app as for the next step of the research

IMPLEMENTATION

Flask API

```python
@app.route( rule: '/predict', methods=['POST'])
def predict():
    try:
        # Get data from the request
        data = request.json
        input_hash = hash(str(data))  # Create a unique hash for the input data

        # Check if a random value is already generated for this input
        if input_hash in random_values:
            random_prediction = random_values[input_hash]
        else:
            # Make predictions
            test_data = pd.DataFrame(data)
            predictions = model_gbc.predict(test_data)

            # Generate a random value within the assigned range
            random_prediction = random.uniform(class_ranges[predictions[0]][0], cla

            # Round the random prediction to the nearest multiple of 10 without dec
            random_prediction = int(round(random_prediction, -1))
```

POSTMAN output



SLIIT
FACULTY OF COMPUTING

IT20017088 | Prabhashi P.A.N. | 2023-133

IMPLEMENTATION

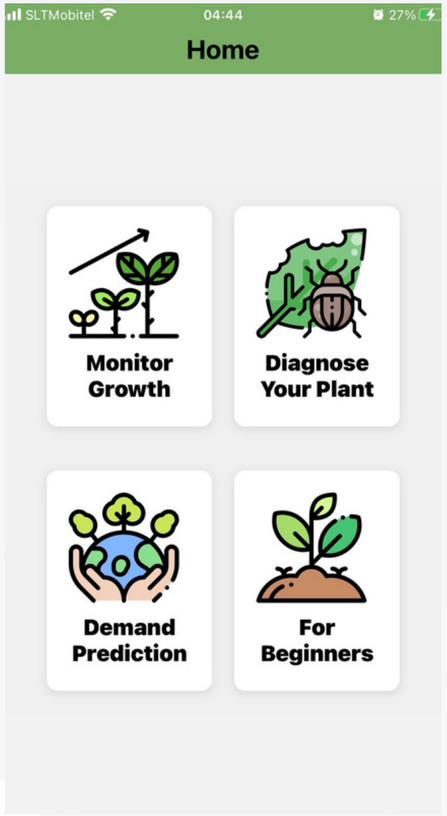
Front end - React Native

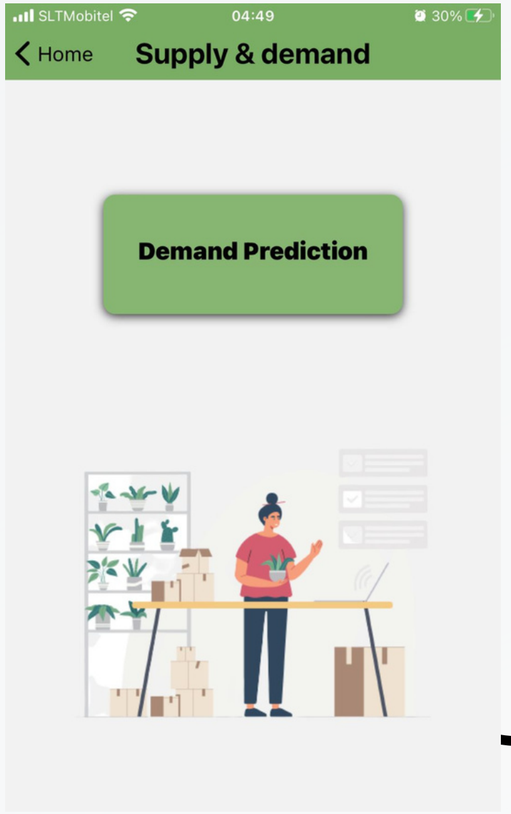
```
import React, { useState } from 'react';
import { View, Text, TextInput, Button, StyleSheet, TouchableHighlight } from 'react-native';
import axios from 'axios';

const DemandSubScreenOne = () => {
  const [prediction, setPrediction] = useState(null);
  const [rainfall, setRainfall] = useState(''); // Default values for testing
  const [temperature, setTemperature] = useState('');
  const [humidity, setHumidity] = useState('');
  const [pot, setPot] = useState('');

  const predict = async () => {
    try {
      const apiUrl = 'http://127.0.0.1/predict';
      const requestBody = {
        Rainfall: [parseFloat(rainfall)],
        Temperature: [parseFloat(temperature)],
        Humidity: [parseFloat(humidity)],
        Pot: [parseFloat(pot)],
      };

      const response = await axios.post(apiUrl, requestBody);

      if (response.status === 200) {
        const responseData = response.data;
        if (responseData.hasOwnProperty('prediction')) {
          setPrediction(responseData.prediction);
        } else {
          throw new Error('Prediction not found in response');
        }
      } else {
        throw new Error('Network response was not ok');
      }
    } catch (error) {
      console.error('Error:', error);
    }
  };

  return (
    <View style={styles.formContainer}>
      <Text style={styles.inputTitle}>Rainfall</Text>
      <TextInput
        placeholder="Rainfall"
        onChangeText={(text) => setRainfall(text)}
        value={rainfall}
        style={styles.inputField}
        keyboardType="numeric"
```

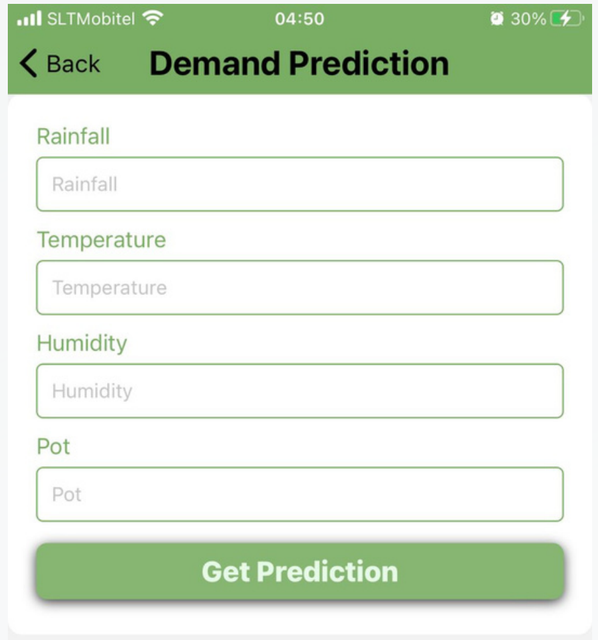
SLIIT FACULTY OF COMPUTING

IT20017088 | Prabhashi P.A.N. | 2023-133

UPDATED UI







SLIIT
FACULTY OF COMPUTING

IT20017088 | Prabhashi P.A.N. | 2023-133

NON-FUNCTIONAL REQUIREMENTS

- Performance with much responsiveness and scalability
- Security
- Reliability
- Usability

FUTURE WORKS

UI Improvement

For better responsive design with the user friendliness. Give a better version of this

App Introducing

Take the feed backs from the users to update the version as wanted and check the outputs frequently and make changes accordingly

IT20005726
Liyanage S.R

GROWTH MONITORING AND PREDICTION SYSTEM FOR ORNAMENTAL PLANTS

SLIIT
FACULTY OF COMPUTING

Specialization : Data Science

RESEARCH PROBLEM

- How can the growth of ornamental plants be monitored, and their current state of growth predicted?

predict a plant's current growth stage based on the observable characteristics it displays at any given time

SLIIT
FACULTY OF COMPUTING

IT20005726 | Liyanage S. R | 2023-133

RESEARCH PROBLEM

- How long will an ornamental plant take to reach the desired growth level for a specific order?

Predictions are based on its current growth stage and the environmental factors provided to the plant

SLIIT
FACULTY OF COMPUTING

IT20005726 | Liyanage S. R | 2023-133

PREVIOUS RESERCHS VS CURRENT APPROACH

Previously

- Expensive smart green house concepts
- IoT based solutions with sensors
- Satellite based solutions
- Not for the Sri Lankan Floriculture Industry

Now

- **Data driven less expensive approach**
- **Based on main characteristics a plant showing at a given time**
- **Consider about the main plants in Sri Lankan Floriculture Industry**

SLIIT
FACULTY OF COMPUTING

IT20005726 | Liyanage S. R | 2023-133

RESEARCH PROGRESS

Then

- Raw data collection
- Not selecting the correct features
- Training the dataset with two models
- Overfitting
- Low accuracy

Now

- Selecting appropriate and correct features after receiving advice from industry experts.
- Implementing proper data cleansing and preprocessing techniques.
- Selecting more suitable models.
- Overcoming overfitting.
- Training the dataset with additional models.
- Creating and testing the backend with the Flask framework.
- Creating the frontend with the React Native app.
- Implementing proper validation for user inputs.
- Simulating with real-world devices.

SLIIT
FACULTY OF COMPUTING

IT20005726 | Liyanage S. R | 2023-133

RESEARCH PROGRESS

Predicting the Growth Category of Ornamental Plants: Growth Monitoring

- Plant Type
- Plant Variety
- Character Leaves
- Average Current Height
- Current Leaf Colour
- Container Size
- Number of Branches
- Dominant Branch
- Growth Duration

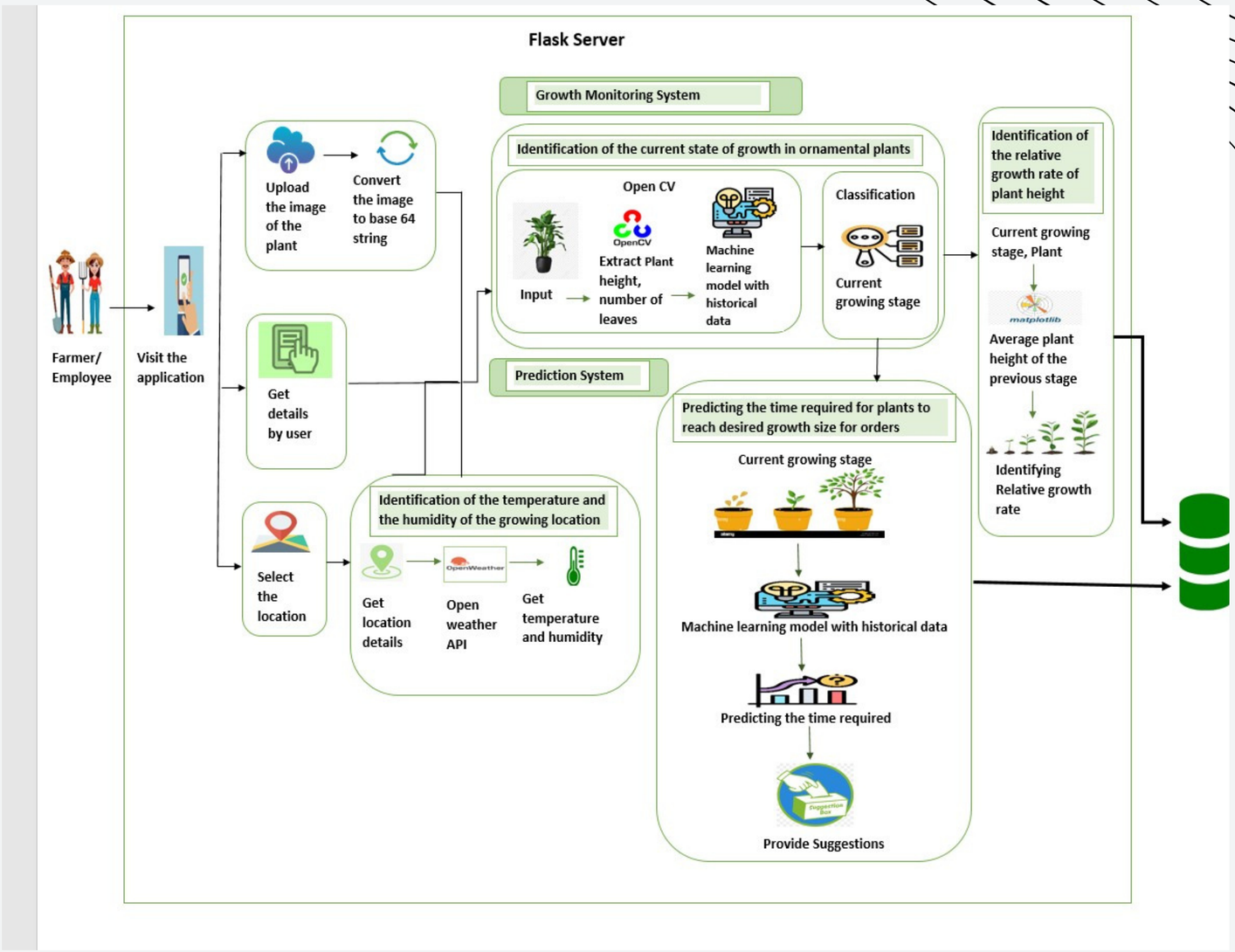
Estimating Time to Reach Desired Growth Level in Ornamental Plants: Growth Duration prediction

- Plant Type
- Plant Variety
- Media Grown
- Fertilizer Type
- Fertilizer Ratio
- Current Growth Category
- Current Pot Type
- Shade Type
- Growth Duration

SLIIT
FACULTY OF COMPUTING

IT20005726 | Liyanage S. R | 2023-133

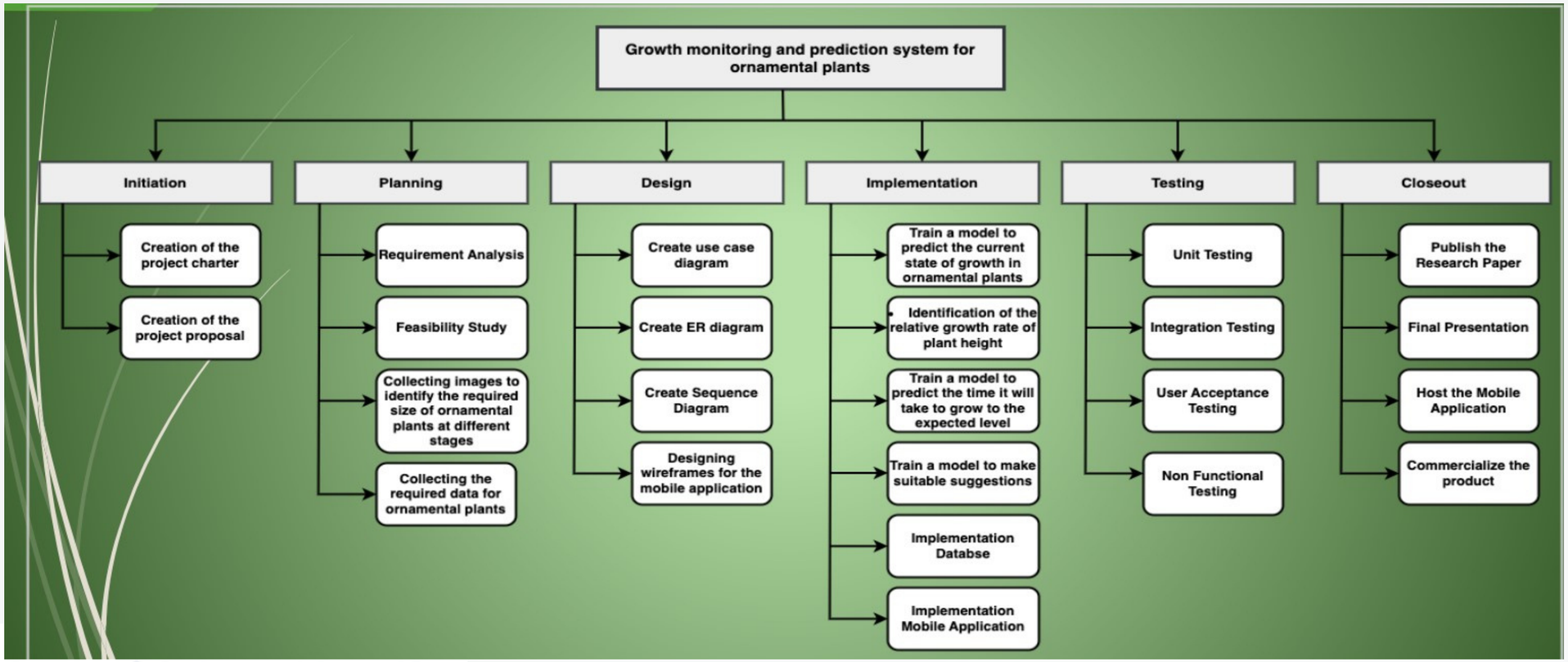
INDIVIDUAL SYSTEM DIAGRAM



SLIIT FACULTY OF COMPUTING

IT20005726 | Liyanage S. R | 2023-133

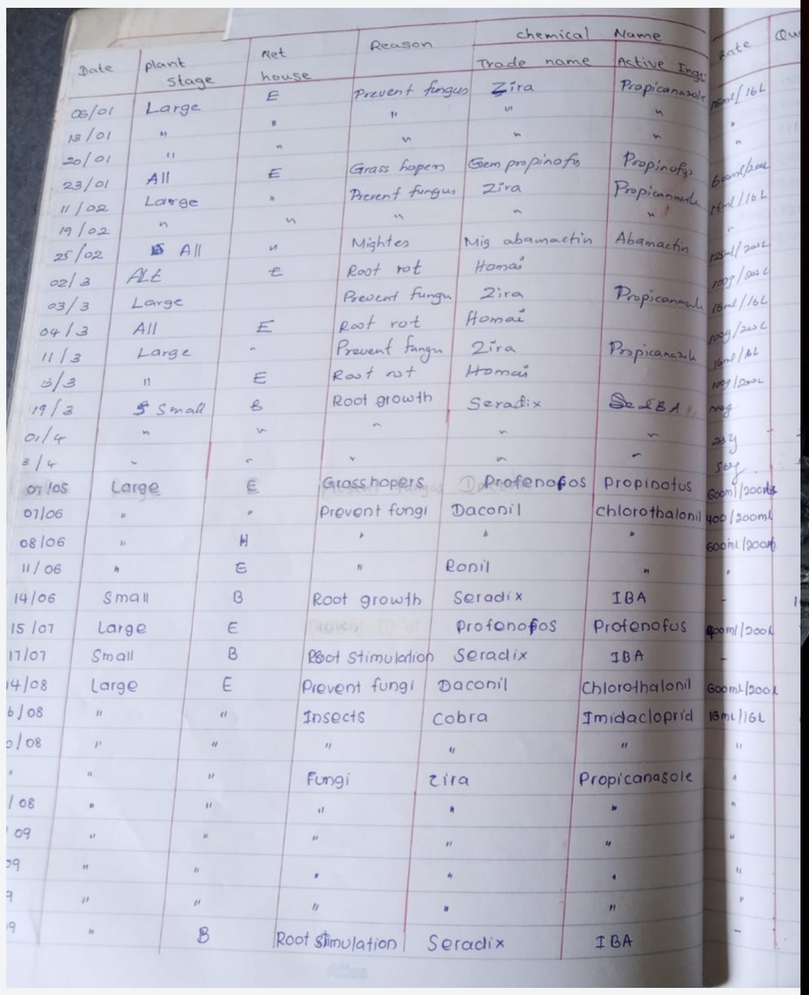
WORK BREAKDOWN



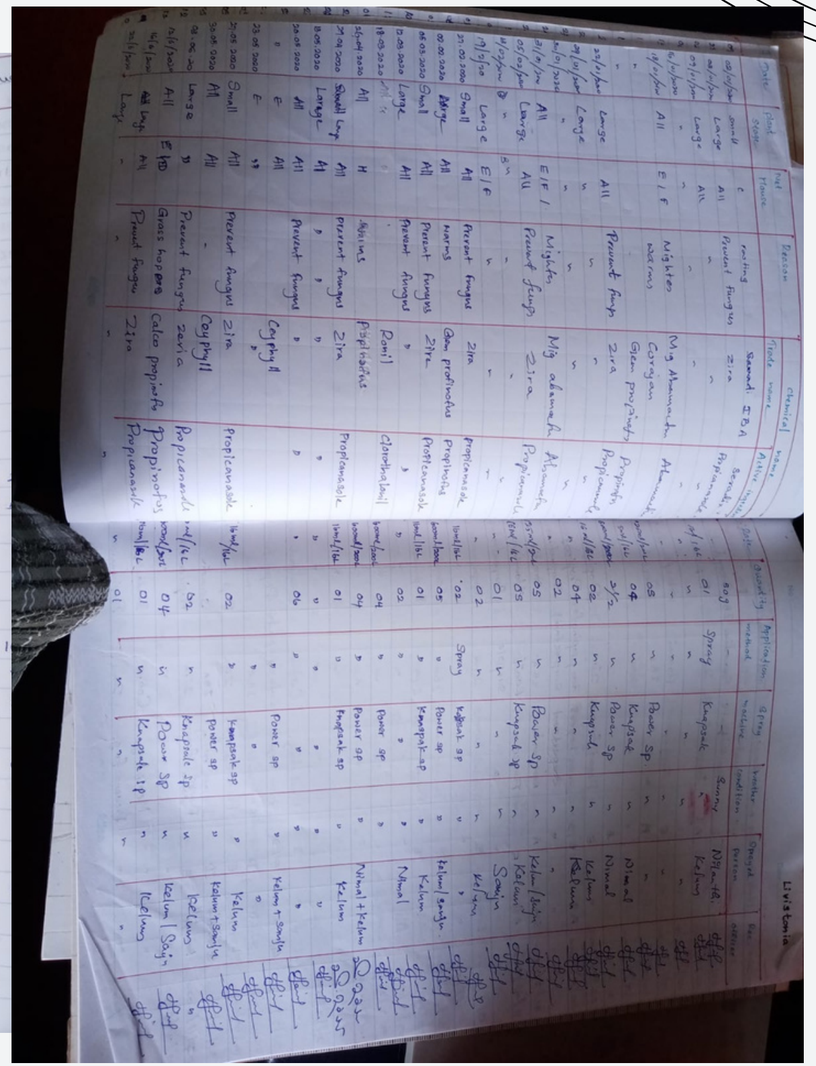
DATA GATHERING

These data is collected from the omega green Pvt Ltd

And all data related to growth monitoring collected in manual way



Date	plant stage	Net house	Reason	chemical Trade name	Name Active Ingr.
06/01	Large	E	Prevent fungus	Zira	Propicanazole
18/01	"	"	"	"	"
20/01	"	"	"	"	"
23/01	All	E	Grass hopers	Gem propinafos	Propinafos
11/02	Large	"	Prevent fungus	Zira	Propicanazole
19/02	"	"	"	"	"
25/02	All	"	Mighter	Mig abamactin	Abamactin
02/3	ALL	E	Root rot	Homai	
03/3	Large		Prevent fungu	Zira	Propicanazole
04/3	All	E	Root rot	Homai	
11/3	Large	"	Prevent fungu	Zira	Propicanazole
13/3	"	E	Root rot	Homai	
19/3	Small	B	Root growth	Seradix	IBA
01/4	"	"	"	"	"
8/4	"	"	"	"	"
07/05	Large	E	Grasshopers	Profenofos	Propinofus
07/06	"	"	Prevent fungi	Daconil	chlorothalonil
08/06	"	H	"	"	"
11/06	"	E	"	Ronil	"
14/06	Small	B	Root growth	Seradix	IBA
15/07	Large	E		Profenofos	Profenofus
17/07	Small	B	Root stimulation	Seradix	IBA
14/08	Large	E	Prevent fungi	Daconil	Chlorothalonil
6/08	"	"	Insects	Cobra	Imidacloprid
0/08	"	"	"	"	"
	"	"	Fungi	Zira	Propicanasole
/08	"	"	"	"	"
09	"	"	"	"	"
09	"	"	"	"	"
9	"	"	"	"	"
9	"	B	Root stimulation	Seradix	IBA



SLIIT FACULTY OF COMPUTING

IT20005726 | Liyanage S. R | 2023-133

DATA GATHERING

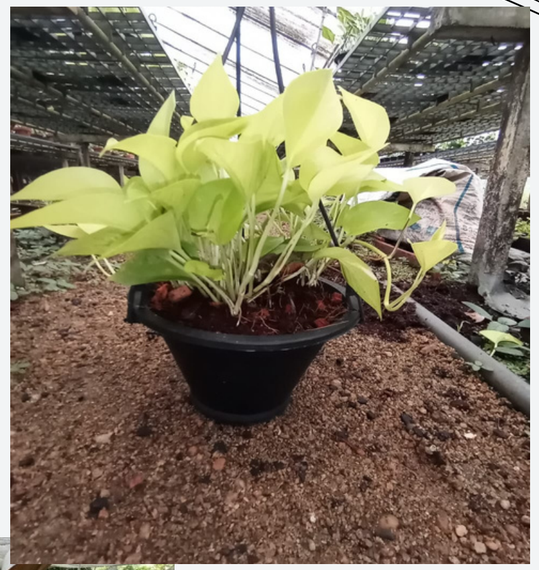
Pictures of main Ornamental plant types considered

Livistonia

Enjoy

Marble Queen

Lemon Lime



Black Cardinal

SLIIT FACULTY OF COMPUTING

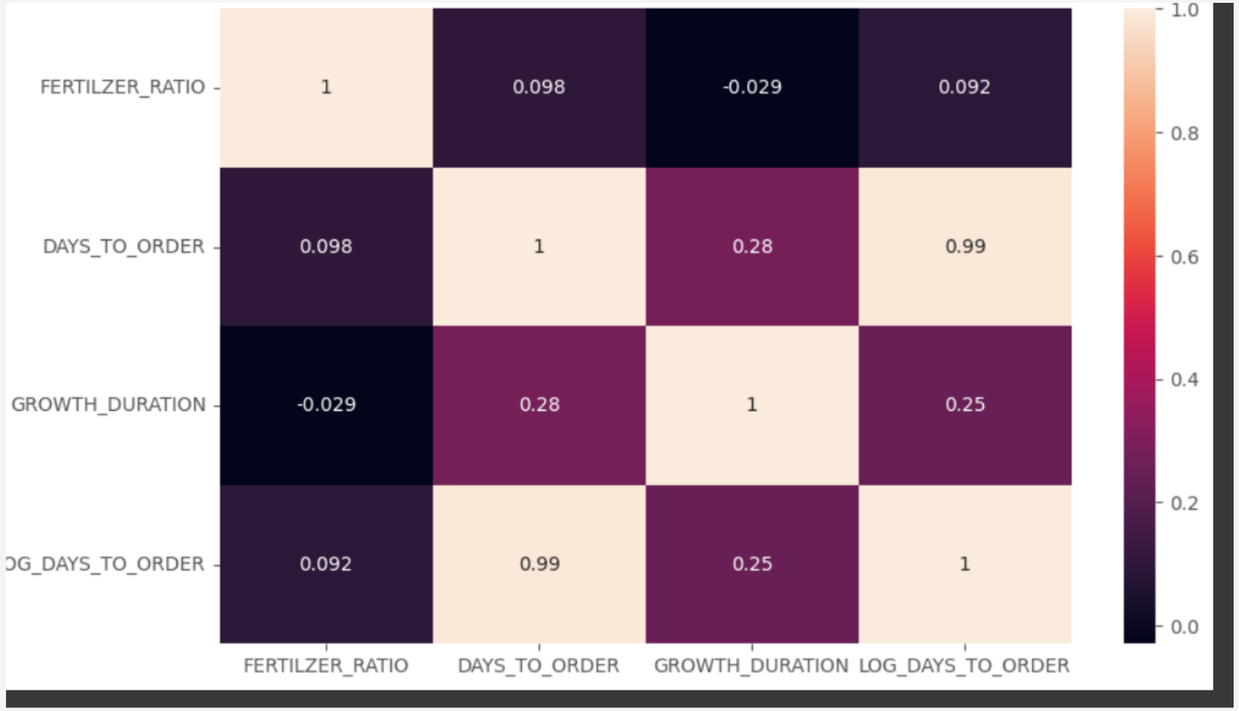
IT20005726 | Liyanage S. R | 2023-133

01.) DATA CLEANSING

- Understanding about the data set
- Handling Null Values
- Duplicates Removing
- Correcting the inconsistency

Handling Numerical missing data

```
[ ]  # Calculate average height for each growth category
     average_heights_by_growth = Growth_Data.groupby('CURRENT_GROWTH_CATEGORY')['AVERAGE_CURRENT_HEIGHT'].mean()

     # Store average heights in variables
     avg_height_mature_weak = average_heights_by_growth['A mature but weak plant']
     avg_height_mature_well = average_heights_by_growth['A mature but well-grown plant']
     avg_height_medium_poor = average_heights_by_growth['A medium-level,poorly growing plant']
     avg_height_medium_well = average_heights_by_growth['A medium-level,well growing plant']
     avg_height_recent_poor = average_heights_by_growth['A recently planted,poorly growing plant']
     avg_height_recent_well = average_heights_by_growth['A recently planted,well growing plant']
     avg_height_small_poor = average_heights_by_growth['A small-level,poorly growing plant']
     avg_height_small_well = average_heights_by_growth['A small-level,well growing plant']

     # Function to impute null values based on growth category average height
     def impute_height(row):
         if pd.isnull(row['AVERAGE_CURRENT_HEIGHT']):
             return average_heights_by_growth[row['CURRENT_GROWTH_CATEGORY']]
         return row['AVERAGE_CURRENT_HEIGHT']

     # Apply the impute_height function to fill null values
     Growth_Data['AVERAGE_CURRENT_HEIGHT'] = Growth_Data.apply(impute_height, axis=1)

     # Check null values after imputation
     null_counts_after = Growth_Data['AVERAGE_CURRENT_HEIGHT'].isnull().sum()
     if null_counts_after == 0:
```



SLIIT
FACULTY OF COMPUTING

IT20005726 | Liyanage S. R | 2023-133

01.) DATA CLEANSING

- Duplicates Removing
- Correcting the inconsistency
- Log transformation

```
[19] import numpy as np

     # Apply log transformation
     Predictions_Data['LOG_DAYS_TO_ORDER'] = np.log1p(Predictions_Data['DAYS_TO_ORDER'])  # log1p is log(1+

     # Plot the transformed variable
     plt.figure(figsize=(10, 5))
     sns.histplot(Predictions_Data['LOG_DAYS_TO_ORDER'], bins=50, kde=True)
     plt.title('Distribution of Log Transformed DAYS_TO_ORDER')
     plt.xlabel('LOG_DAYS_TO_ORDER')
     plt.ylabel('Frequency')
     plt.show()
```

adding meaningful attribute values to maintain the consistency

```
[12] # List of columns to preprocess. These are mainly categorical columns which might have inconsistent string formats.
     columns_to_preprocess = ['PLANT_TYPE', 'MEDIA_GROWN','FERTILZER_TYPE' ,'CURRENT_POT_TYPE', 'CURRENT_LIFE_TIME', 'CURRENT_LEAF_COLOUR', 'CONTAINER_SIZE','SHADE_TYPE']

     for column in columns_to_preprocess:
         # Replacing spaces with underscores for consistency.
         # It helps avoid potential issues with databases or other systems that might misinterpret spaces.
         Predictions_Data[column] = Predictions_Data[column].str.replace(' ', '_')

         # Convert all string data to lowercase to ensure uniformity.
         # This can prevent potential mismatches due to case differences.
         Predictions_Data[column] = Predictions_Data[column].str.lower()

         # Trim any leading or trailing spaces from the strings.
         # These spaces can cause inconsistencies and often go unnoticed.
         Predictions_Data[column] = Predictions_Data[column].str.strip()
         # Add more preprocessing steps as needed

     # After preprocessing, it's useful to inspect unique values of these columns.
     # This helps in identifying any remaining inconsistencies or anomalies in the data.
     for column in columns_to_preprocess:
         print(f"Unique values in {column}:")
         print(Predictions_Data[column].unique())
         print("\n")

     Unique values in PLANT_TYPE:
```

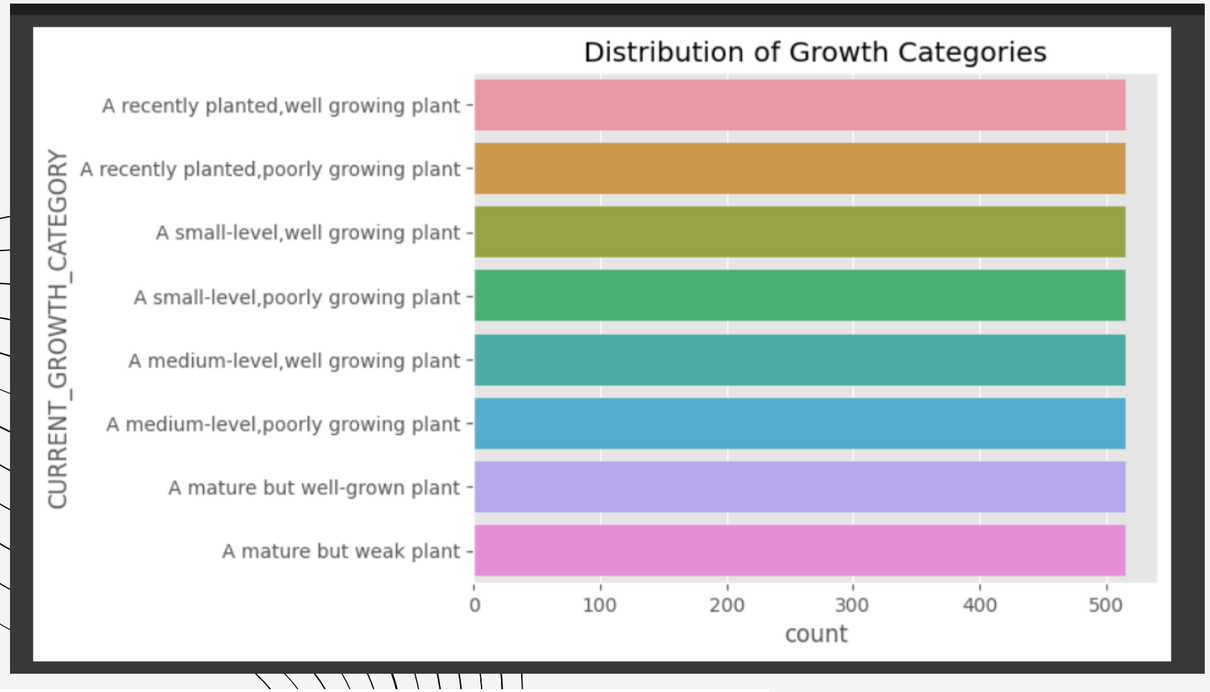
SLIIT
FACULTY OF COMPUTING

IT20005726 | Liyanage S. R | 2023-133

02.)UNDERSTANDING ABOUT THE DISTRIBUTION

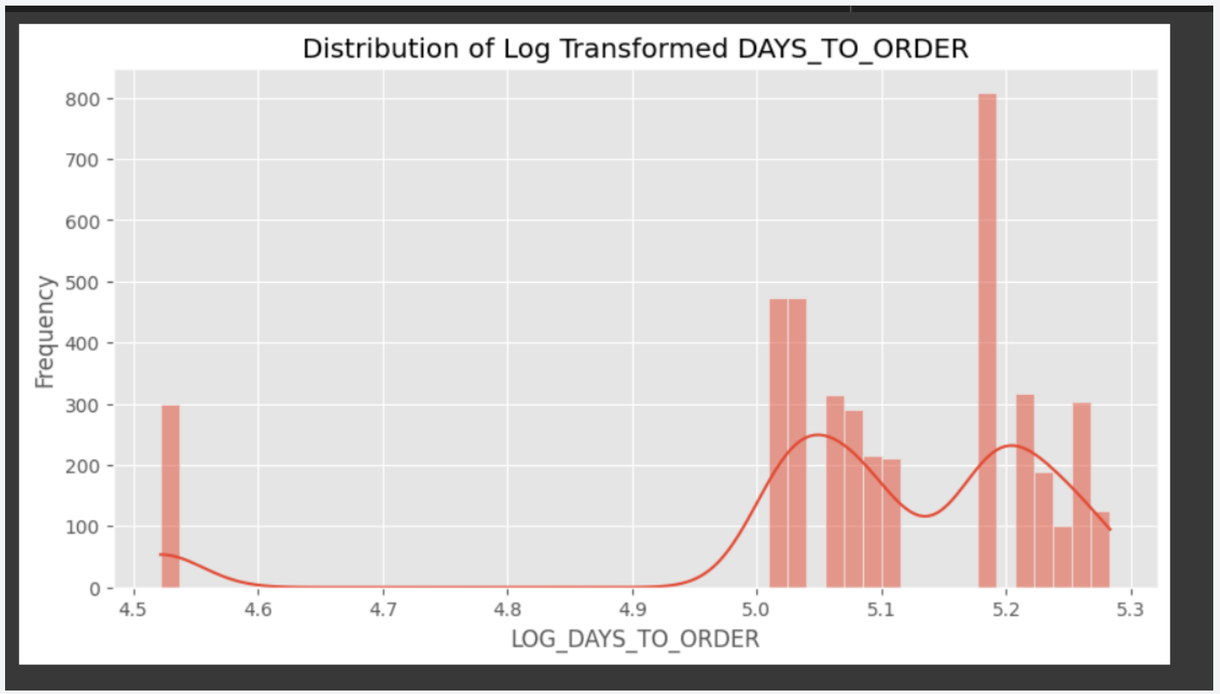
Growth Monitoring

Distribution Of Growth Categories



Growth Duration prediction

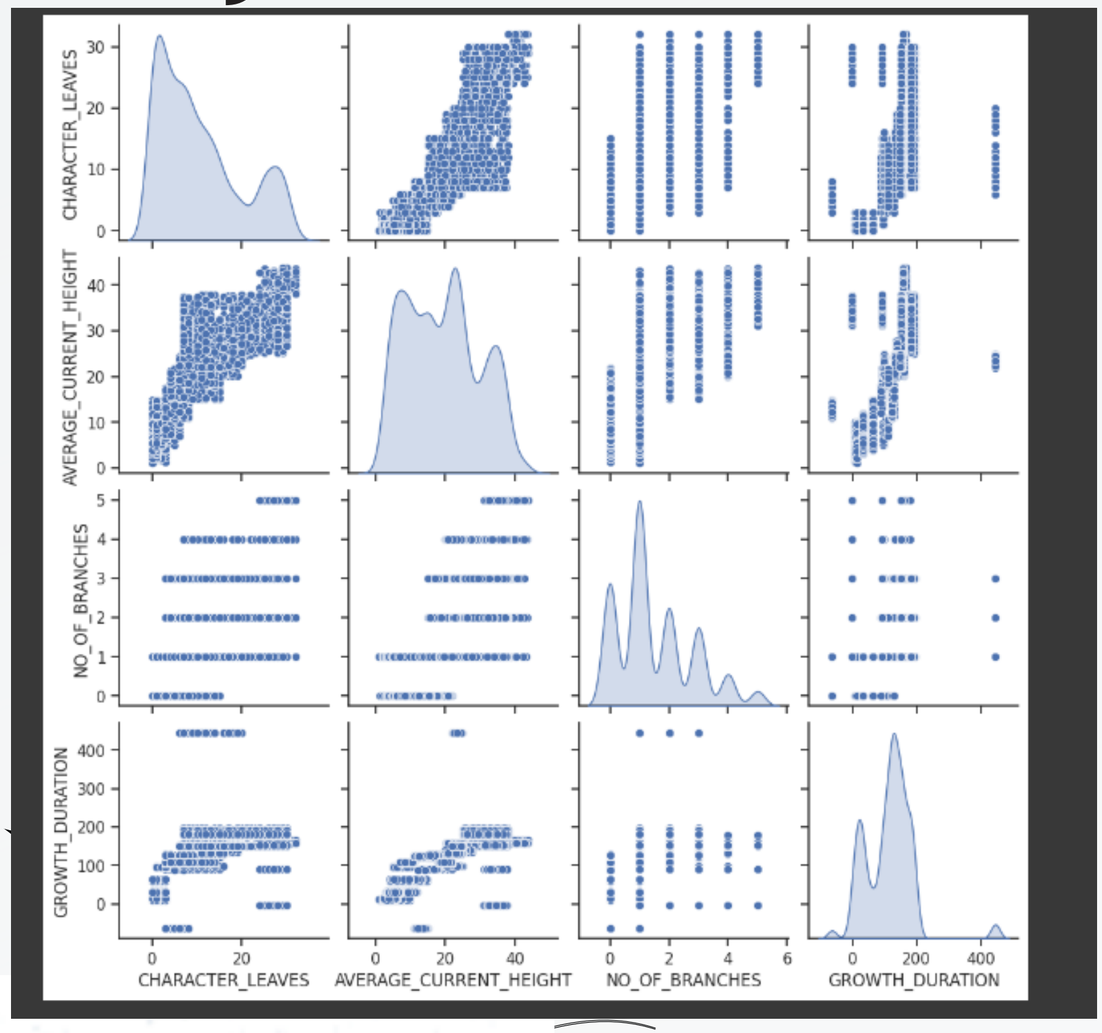
Distribution Of Growth Duration



SLIIT
FACULTY OF COMPUTING

IT20005726 | Liyanage S. R | 2023-133

03.)DATA VISUALIZATION



Visualizing Categorical and Numerical Data

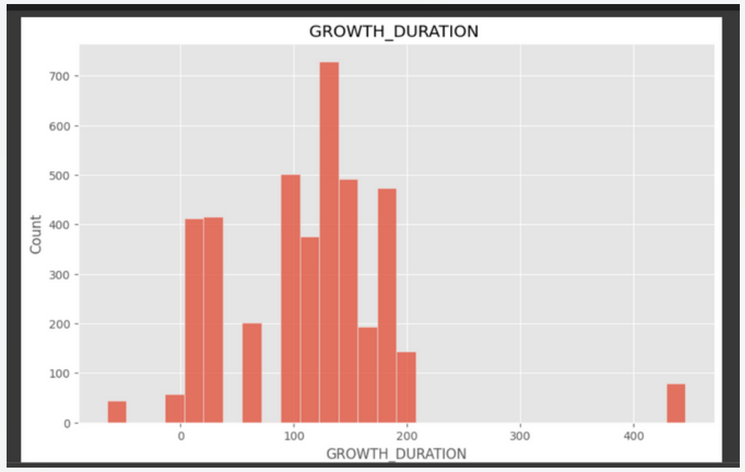
Pair Plots

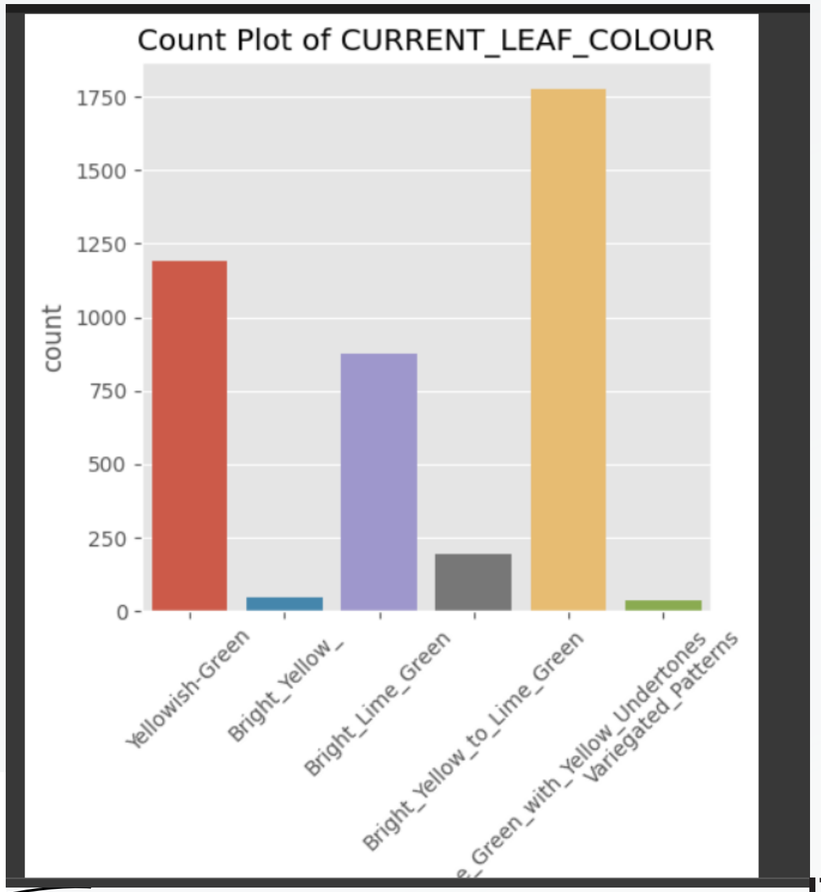
SLIIT
FACULTY OF COMPUTING

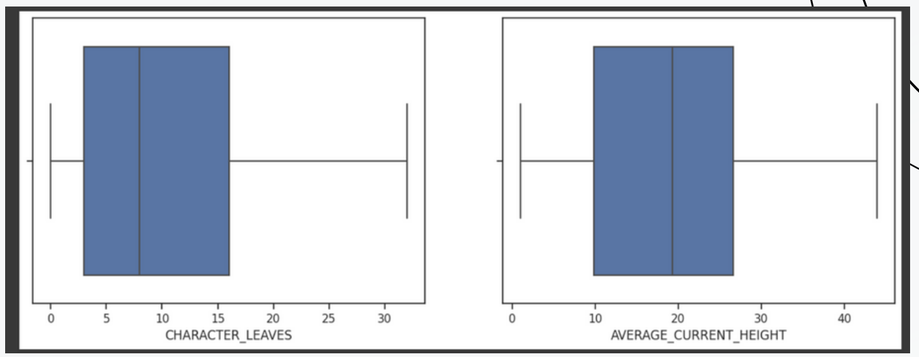
IT20005726 | Liyanage S. R | 2023-133

03.)DATA VISUALIZATION

Visualizing Categorical and Numerical Data







Bar Charts

Box Plots

SLIIT FACULTY OF COMPUTING

IT20005726 | Liyanage S. R | 2023-133

04.)MODEL CREATION

Growth Monitoring

- **Multiclass Classification Problem**
- **Identifying the target variable**
- **Categorical and Numerical data**
- **Avoid dummy variable trap**

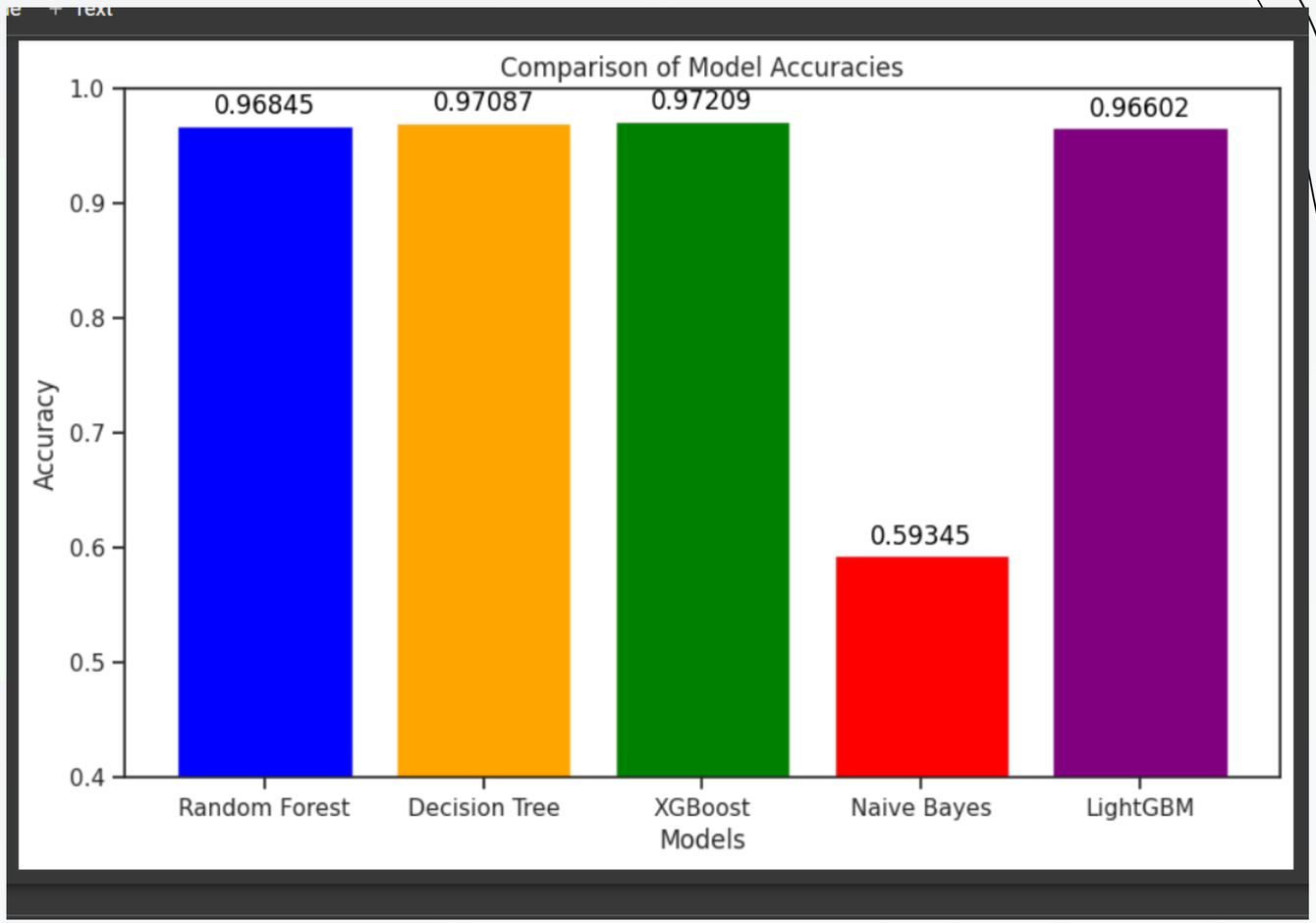
Chosen Models:

- Random Forest Classifier
- Decision Tree Classifier
- XGBoost Classifier
- Naïve Bayes Classifier
- LightGBM Classifier

MODEL COMPARRISON

Growth Monitoring

- Random Forest Classifier
- Decision Tree Classifier
- XGBoost Classifier
- Naïve Bayes Classifier
- LightGBM Classifier



SLIIT
FACULTY OF COMPUTING

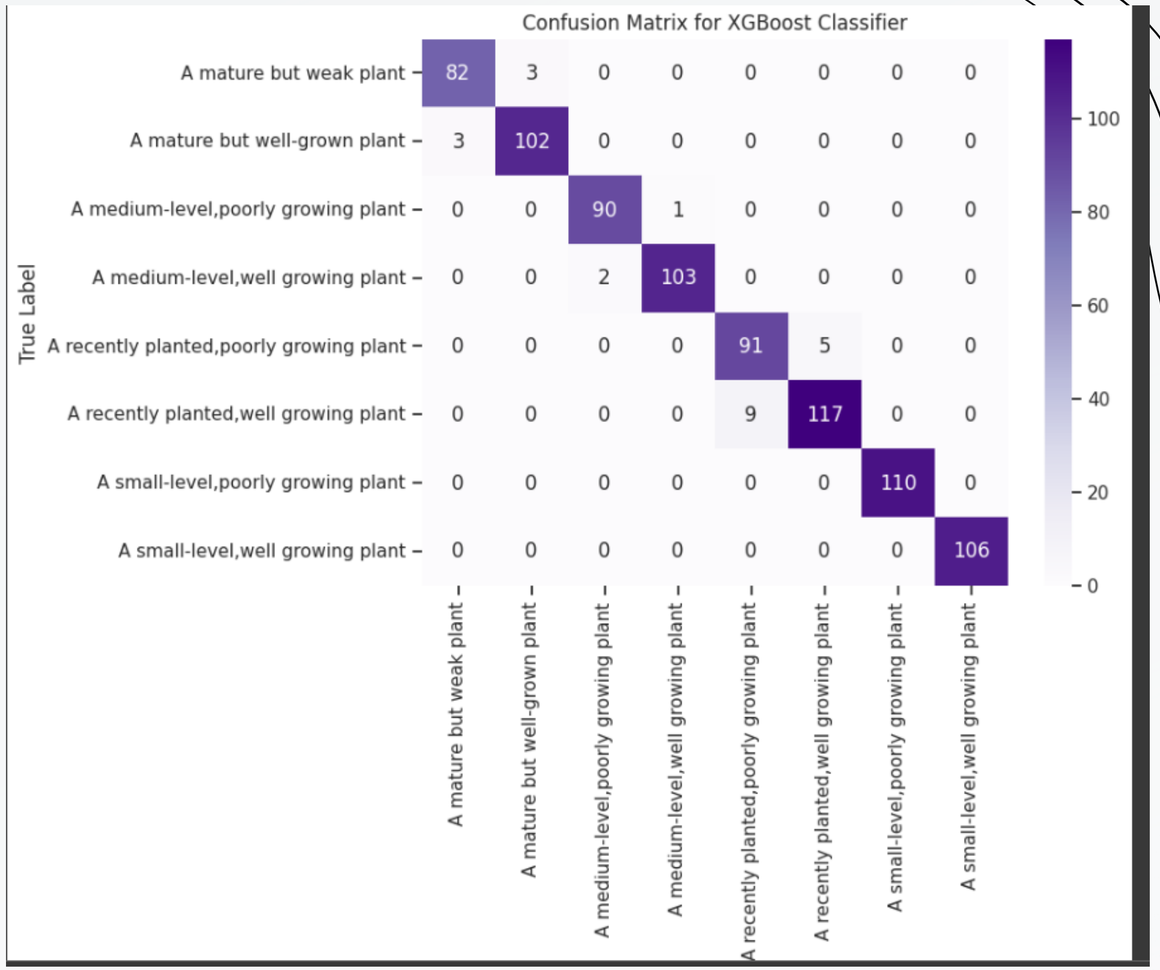
IT20005726 | Liyanage S. R | 2023-133

MODEL CREATION

Growth Monitoring--Model Selection

XGBoost Classifier

- Accuracy = 97.02%
- Gradient Boosting Mechanism
- Automatic Handling of Missing Data
- Built-in Regularization
- Scalability with Large Datasets
- Tree Pruning
- Cross-validation Capabilities



SLIIT FACULTY OF COMPUTING

IT20005726 | Liyanage S. R | 2023-133

MODEL CREATION

Predicting the growth time

- No need for stationarity.
- Simplicity in implementation and interpretation.
- Capability to handle multiple independent variables.
- Absence of a consistent temporal component.
- Less computational intensity.
- Demonstrated predictive accuracy.
- Emphasis on feature influence over sequential data.
- No evident cyclic or seasonal patterns.

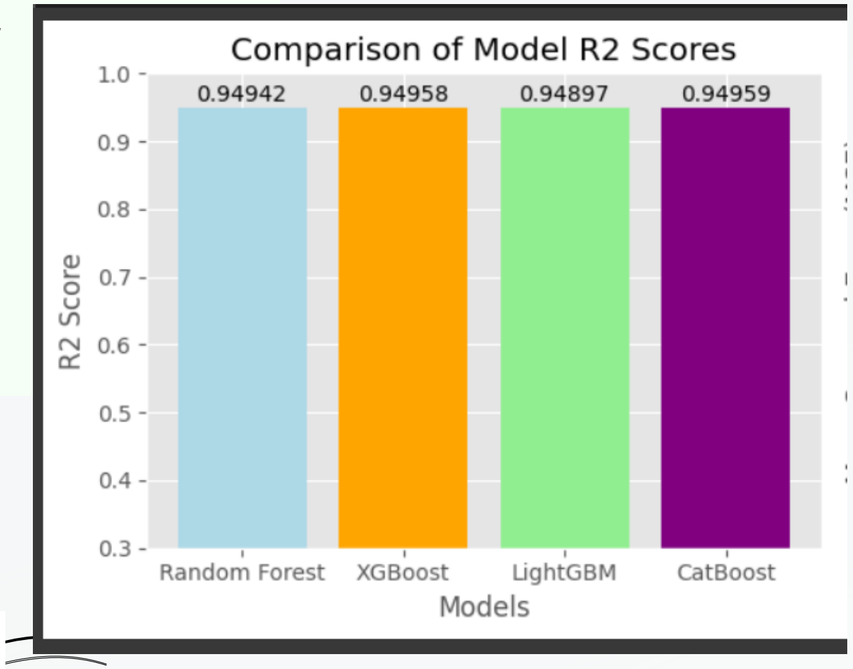
- Random Forest Regressor
- Decision Tree Regressor
- XGBoost Regressor
- CatBoost Regressor

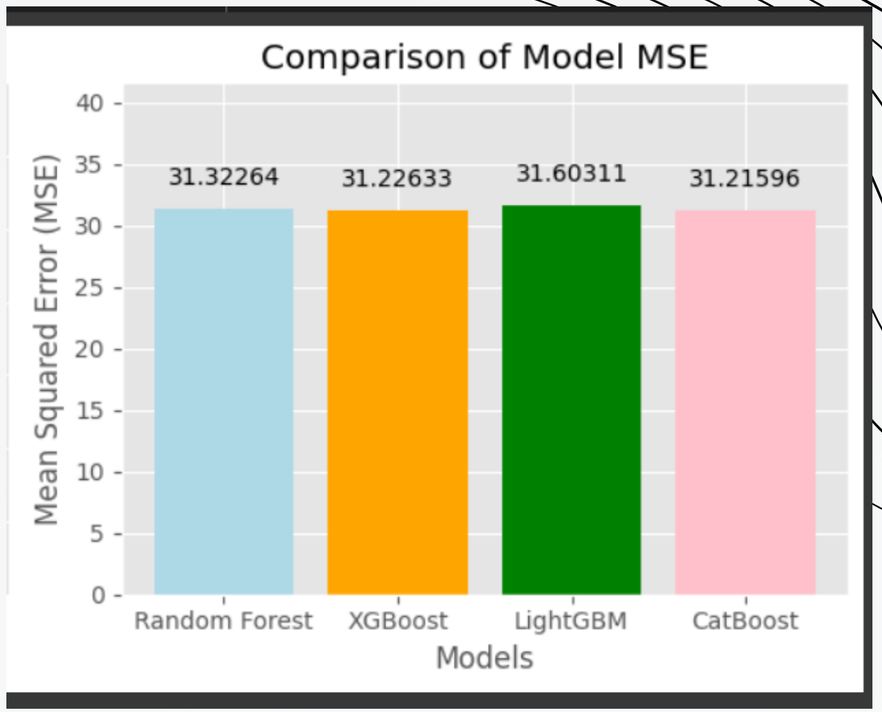
MODEL CREATION

Predicting the growth time

- Random Forest Regressor
- Decision Tree Regressor
- XGBoost Regressor
- CatBoost Regressor

R2 score Comparison





MSE Comparison

SLIIT
FACULTY OF COMPUTING

IT20005726 | Liyanage S. R | 2023-133

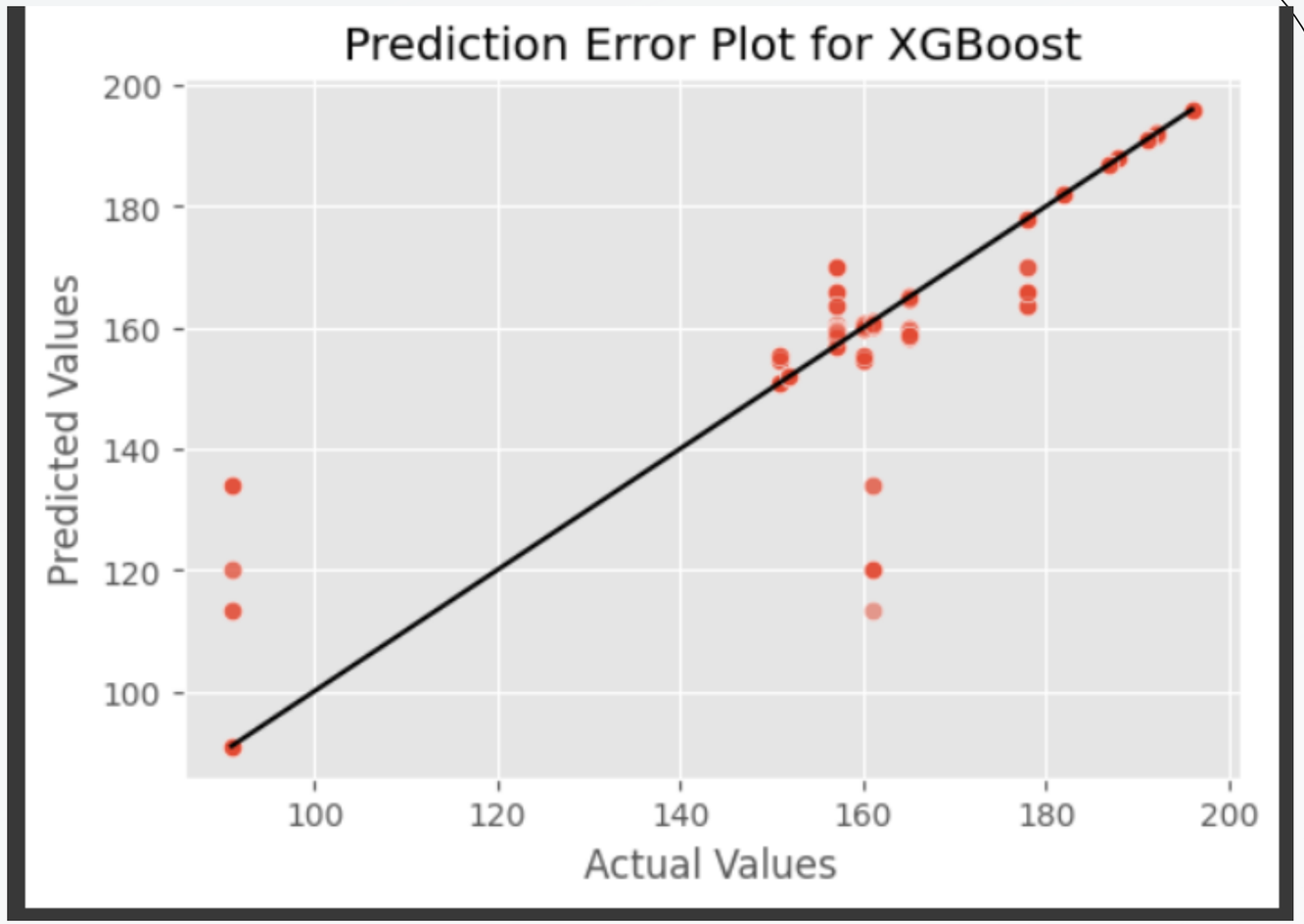
MODEL CREATION

Predicting the growth Duration

XGBoost Regressor

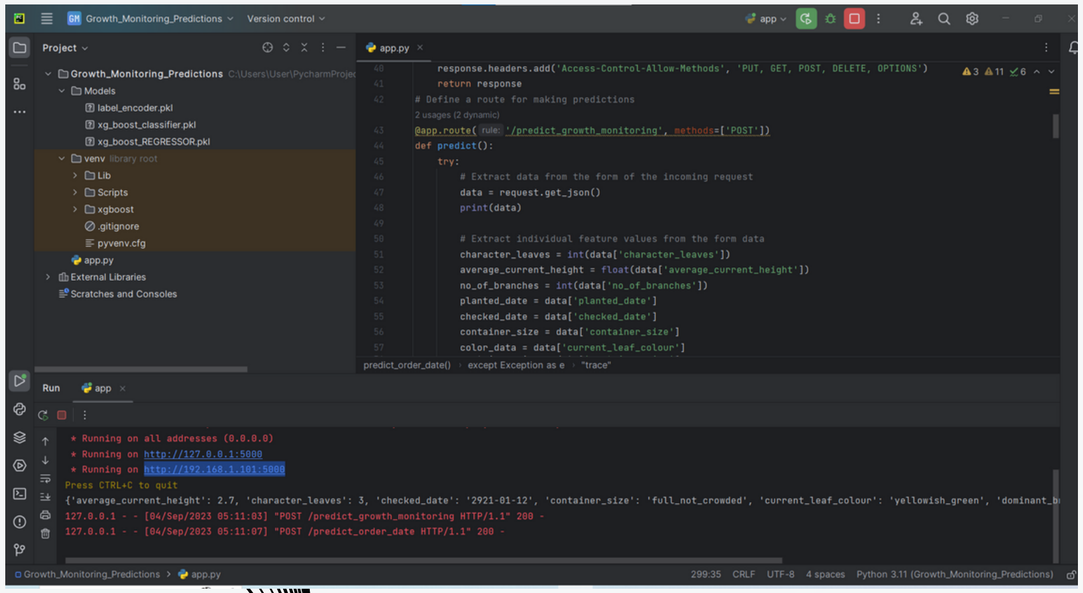
- **R2 score = 94%**
- **MSE**

Prediction Error Plot

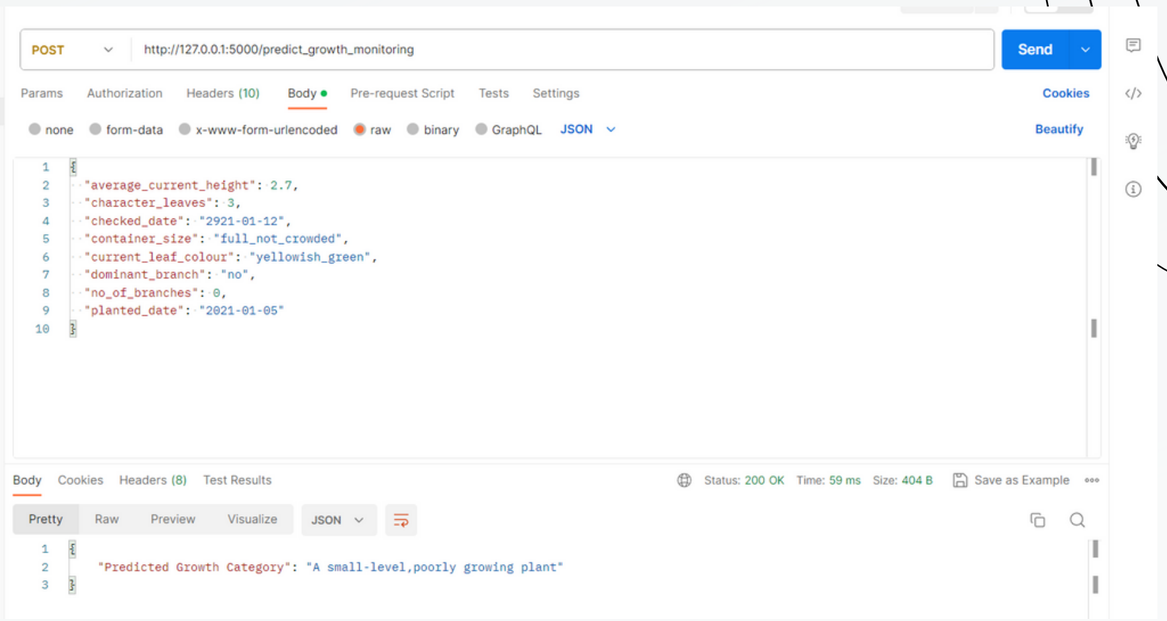


05.)BACKEND IMPLEMENTATION AND TESTING

Growth Monitoring



Backend creation with Flask Framework



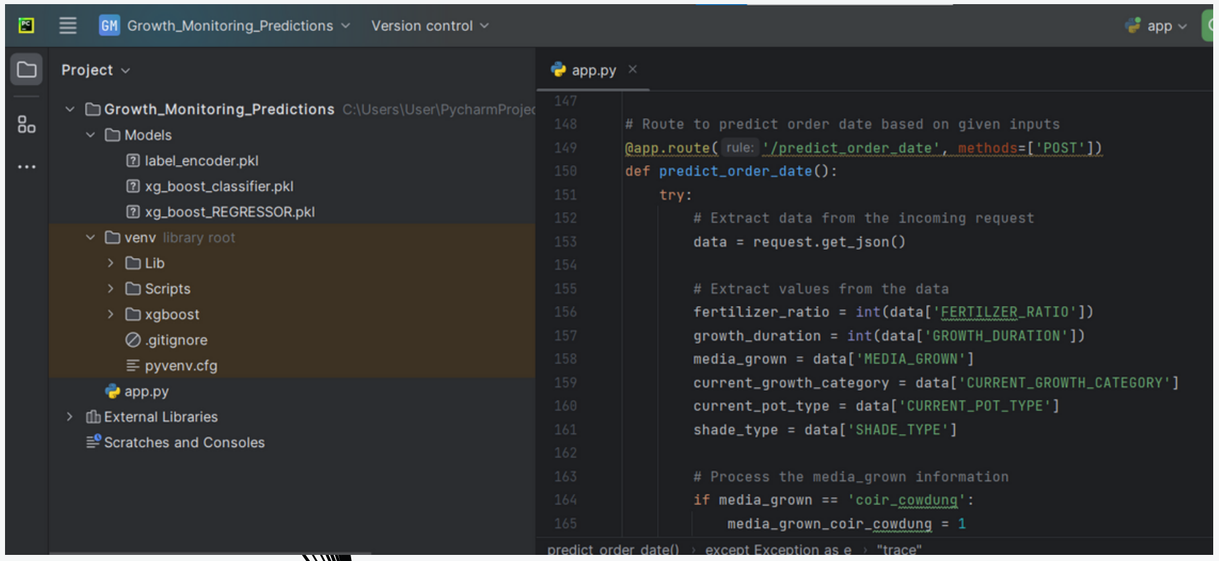
Backend testing with postman

SLIIT
FACULTY OF COMPUTING

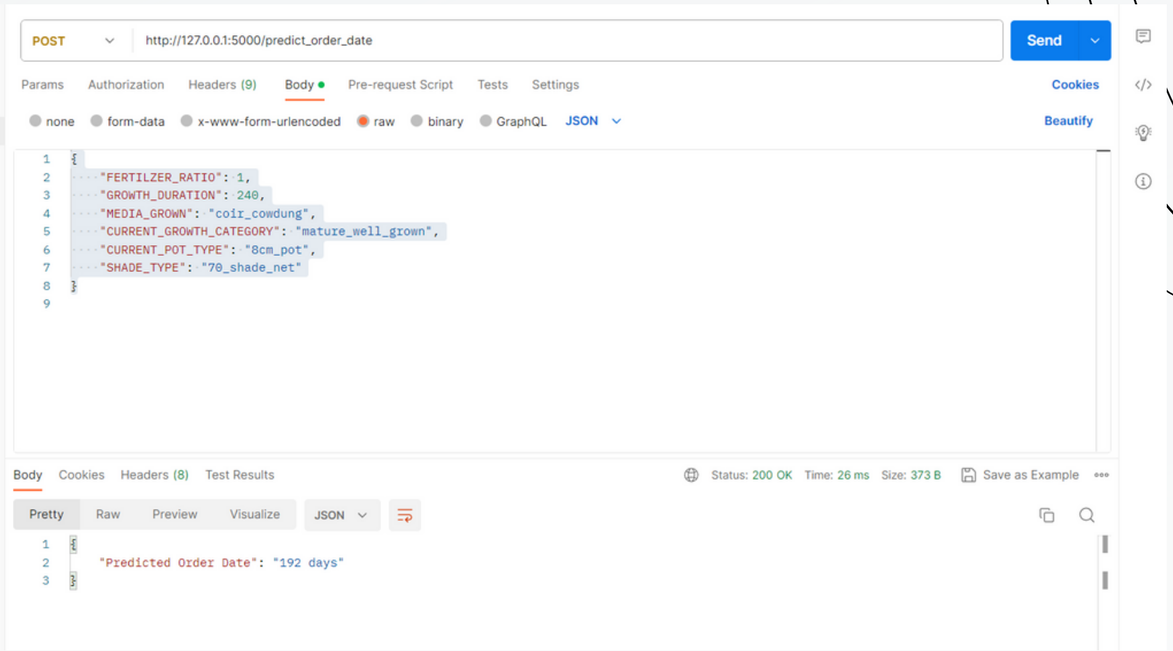
IT20005726 | Liyanage S. R | 2023-133

05.)BACKEND IMPLEMENTATION AND TESTING

Growth Duration Prediction



Backend creation with Flask Framework



Backend testing with postman

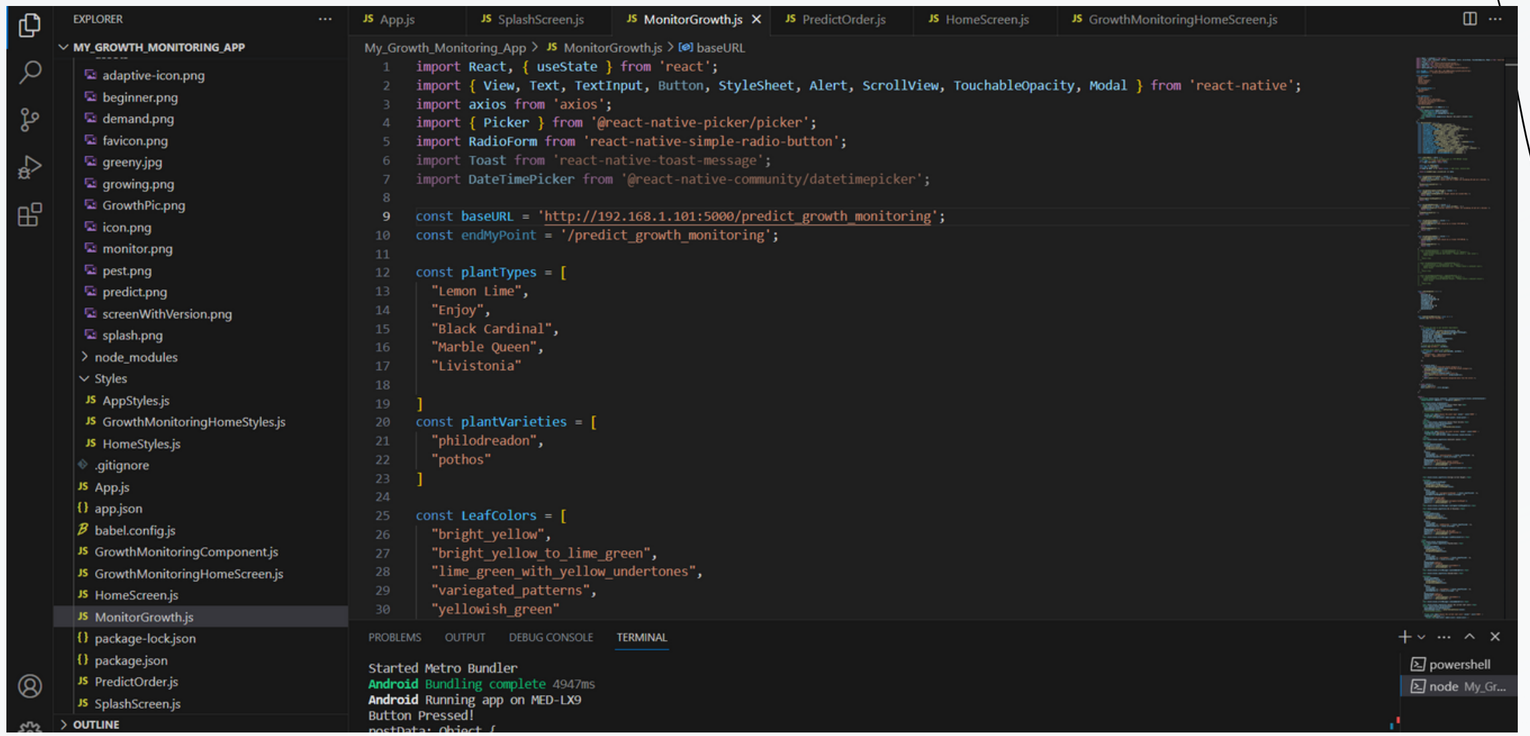
SLIIT
FACULTY OF COMPUTING

IT20005726 | Liyanage S. R | 2023-133

06.)FRONTEND CREATION

Growth Monitoring

Frontend creation with React native



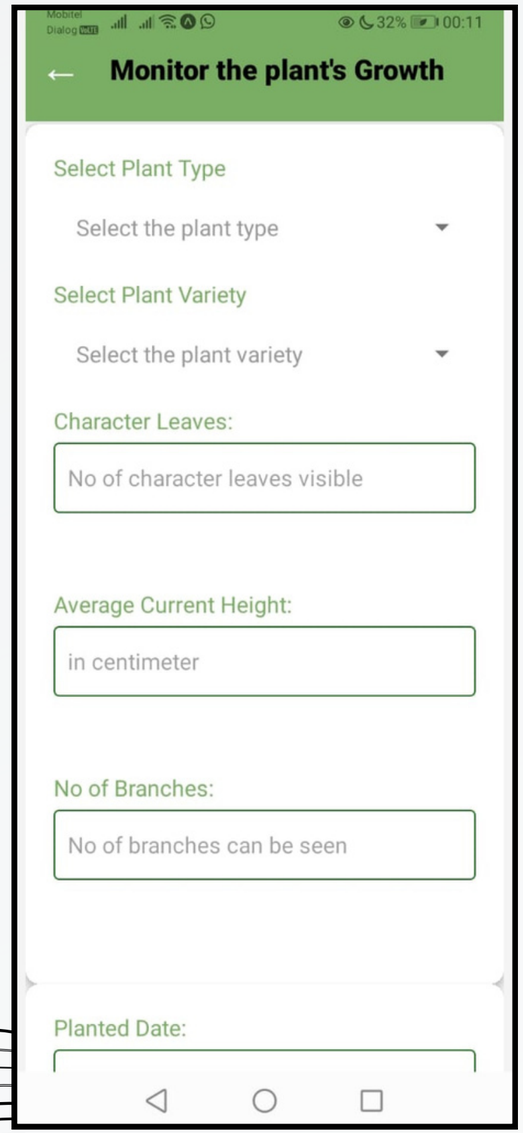
SLIIT
FACULTY OF COMPUTING

IT20005726 | Liyanage S. R | 2023-133

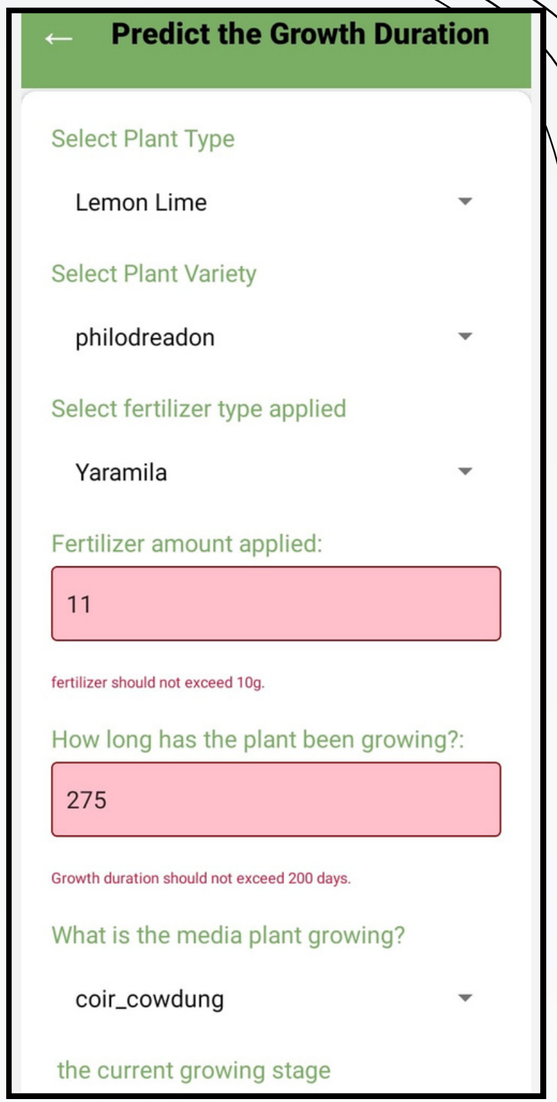
06.)FRONTEND CREATION

Growth Monitoring

Collecting user input on the plant's current observable characteristics



Validating the user's input



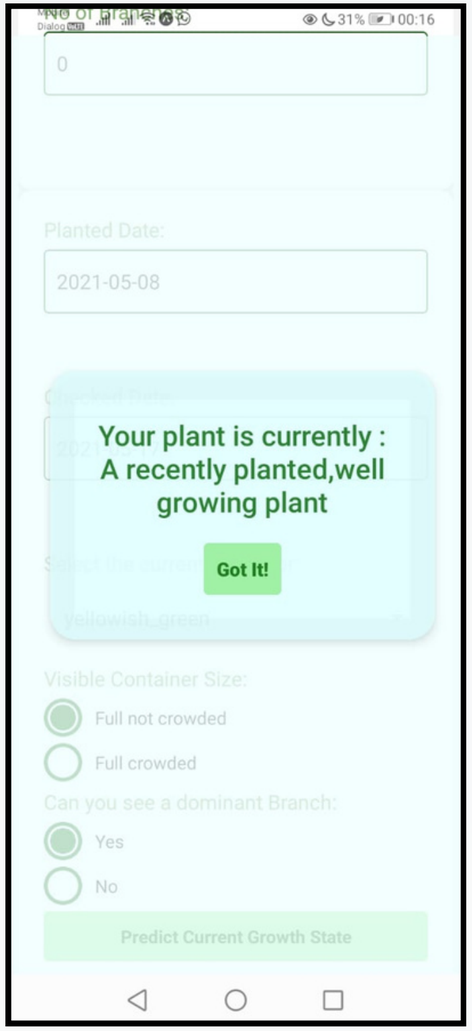
SLIIT FACULTY OF COMPUTING

IT20005726 | Liyanage S. R | 2023-133

06.)FRONTEND CREATION

Growth Monitoring

Identifying the plant's growth category based on the given characteristics



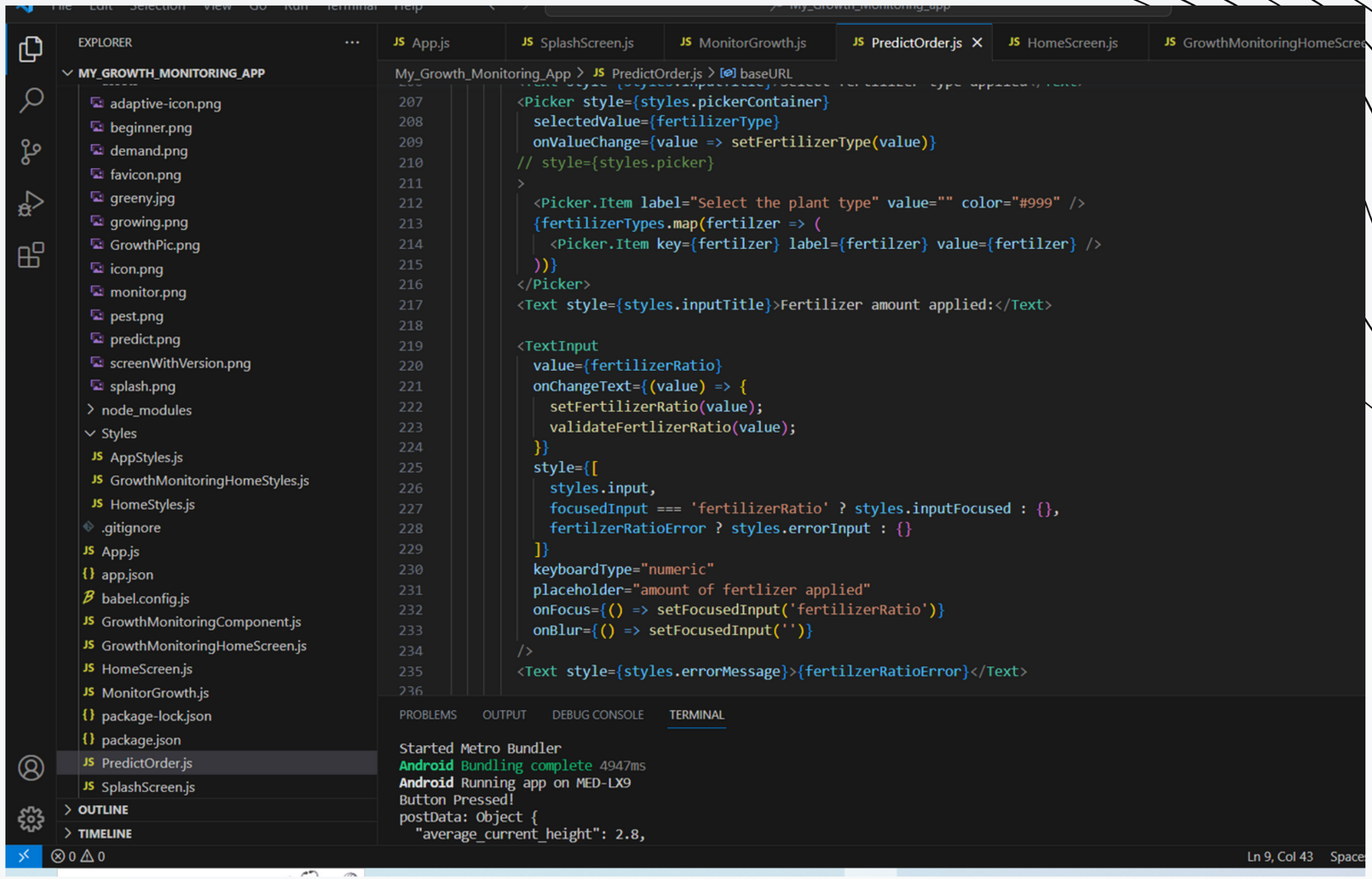
SLIIT
FACULTY OF COMPUTING

IT20005726 | Liyanage S. R | 2023-133

06.)FRONTEND CREATION

Growth Duration Prediction

Frontend creation with React native



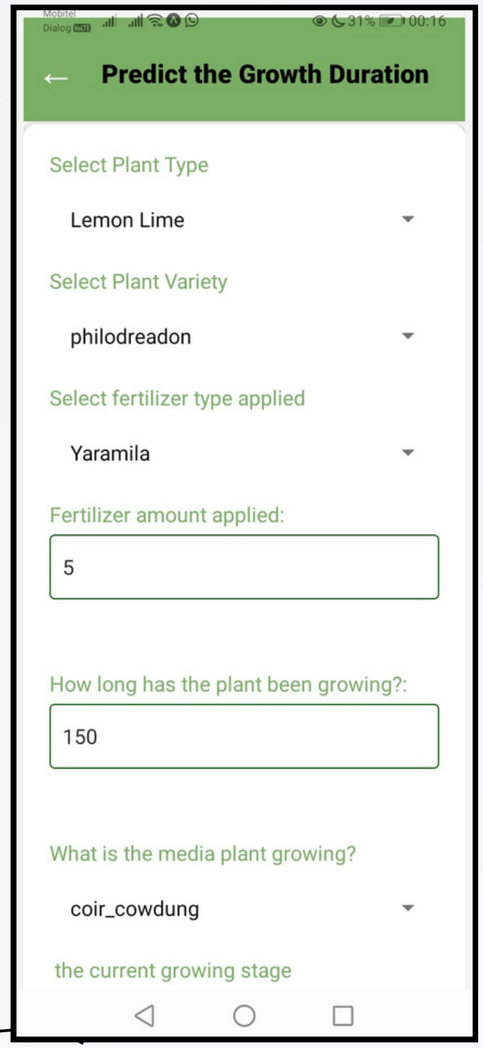
SLIIT
FACULTY OF COMPUTING

IT20005726 | Liyanage S. R | 2023-133

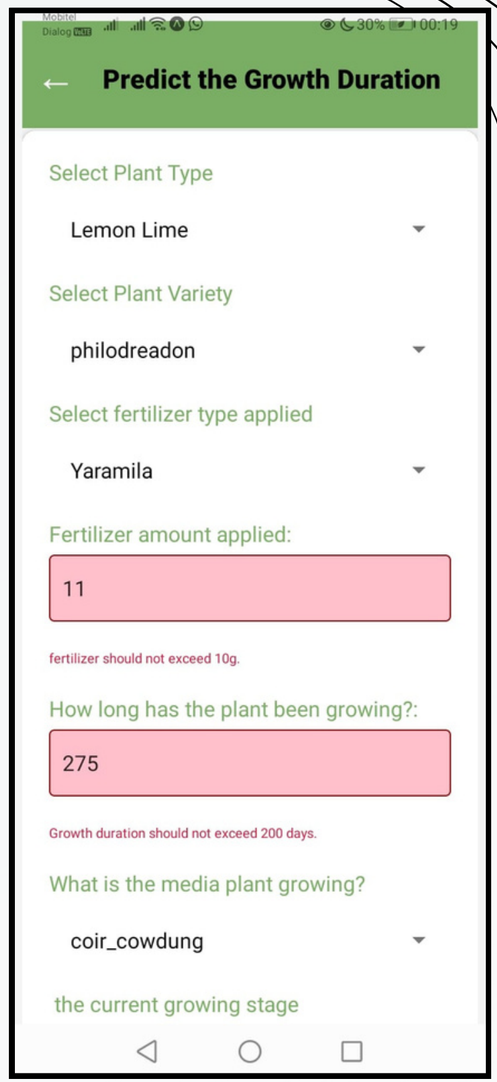
06.)FRONTEND CREATION

Growth Duration Prediction

Gathering information on the plant's current growth stage and its environmental conditions.



Validating the user's input



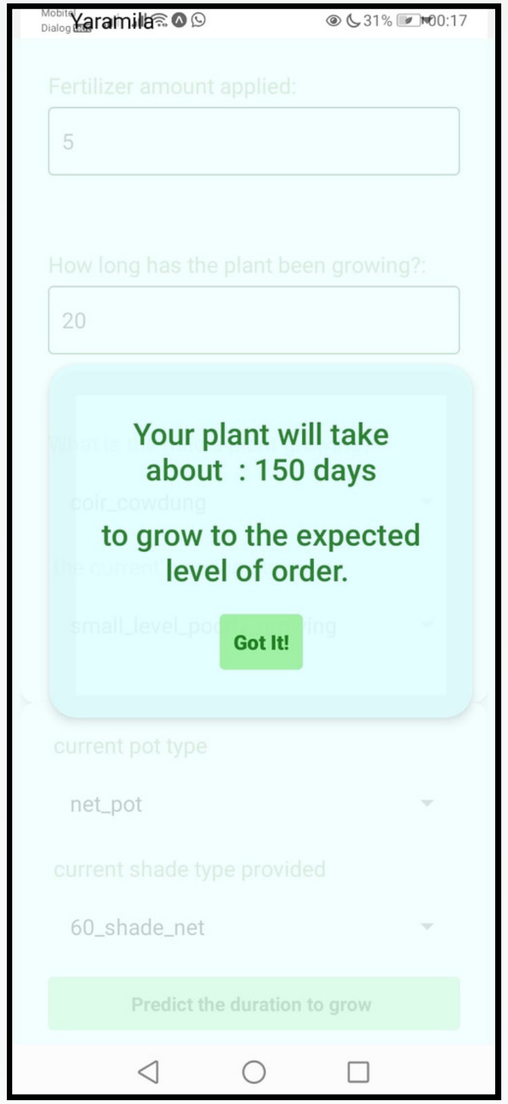
SLIIT FACULTY OF COMPUTING

IT20005726 | Liyanage S. R | 2023-133

06.)FRONTEND CREATION

Growth Duration Prediction

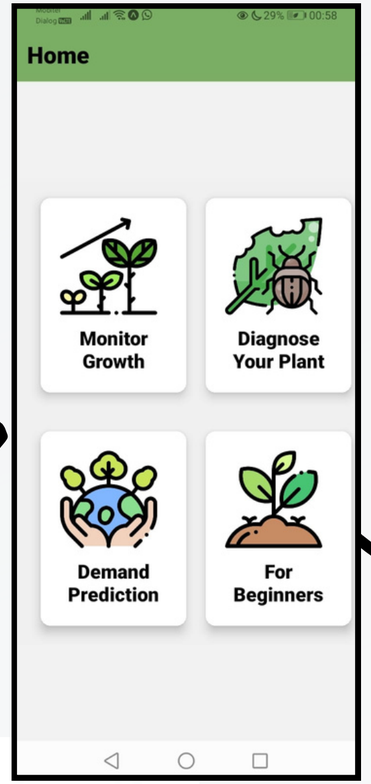
Estimating the duration for the plant to reach the desired growth level for an order

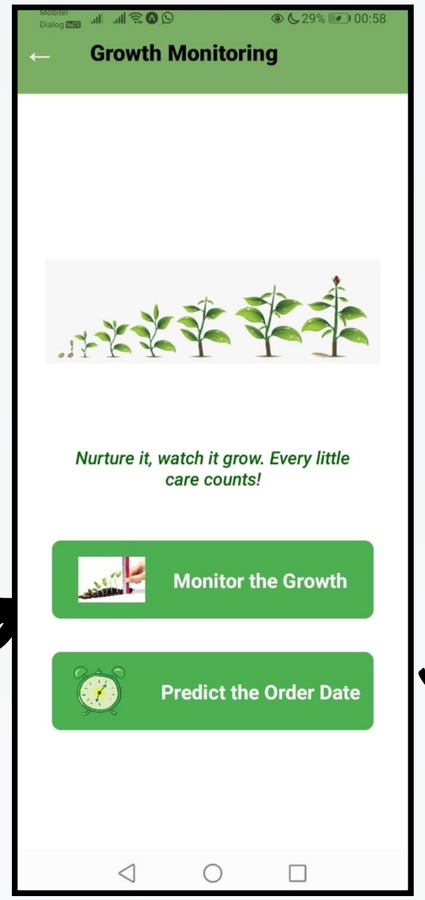


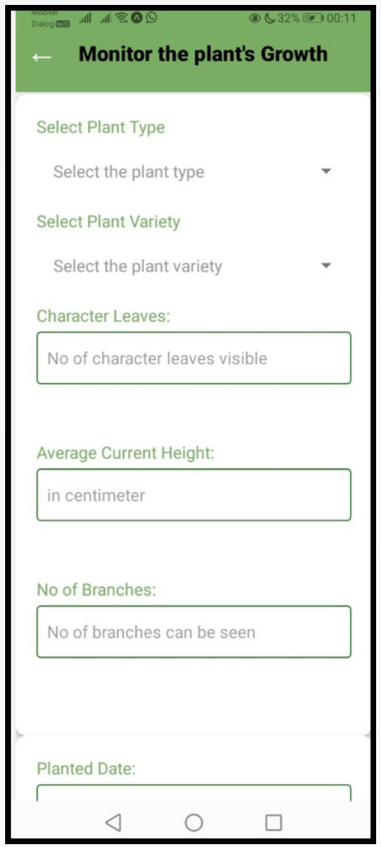
SLIIT FACULTY OF COMPUTING

IT20005726 | Liyanage S. R | 2023-133

UPDATED UI







SLIIT
FACULTY OF COMPUTING

IT20005726 | Liyanage S. R | 2023-133

NON-FUNCTIONAL REQUIREMENTS

- User-friendly and visually appealing interface
- Responsive and provides results in a timely manner
- Compatible with both iOS and Android operating systems
- Accurate and provides reliable predictions
- Fast performance

FUTURE WORKS

Enhancement of User Interfaces:

aim to refine and optimize the design and functionality of our user interfaces to offer a more intuitive and engaging experience for users.

Advancements in Model Precision:

: Continuous efforts will be dedicated to further research and fine-tuning, with the goal of enhancing the predictive accuracy of our model. This involves exploring more advanced algorithms, feature engineering, and data augmentation techniques.

SLIIT
FACULTY OF COMPUTING

IT20005726 | Liyanage S. R | 2023-133

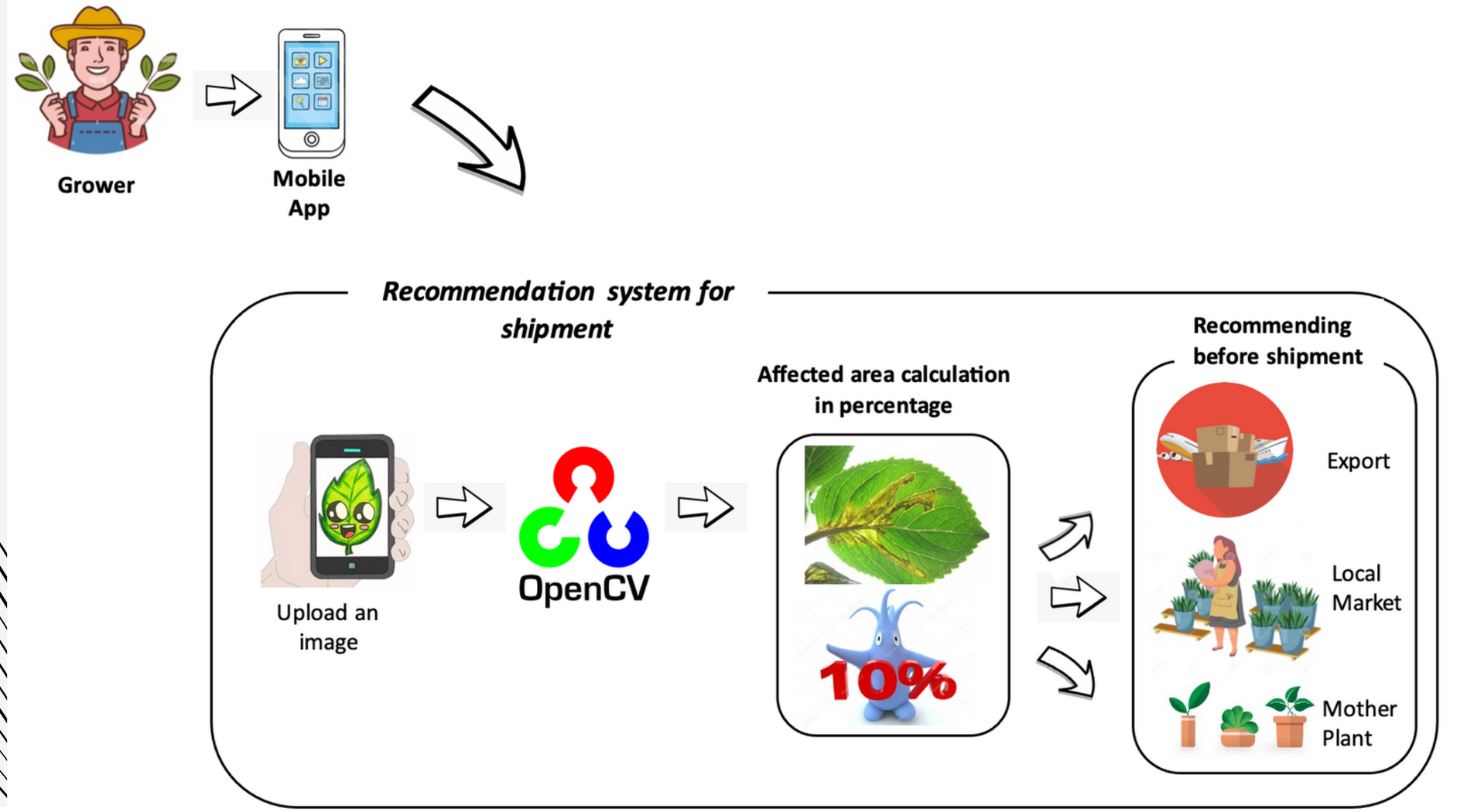
IT19994406
Basnayake N.S.N.

LEAF AFFLICTION ANALYSIS AND SHIPMENT RECOMMENDATION SYSTEM

SLIIT
FACULTY OF COMPUTING

Specialization : Data Science

INDIVIDUAL SYSTEM DIAGRAM



RESEARCH PROBLEM

How to determine whether to,

- Export leaves
- Sell them locally
- Keep them for the mother plant

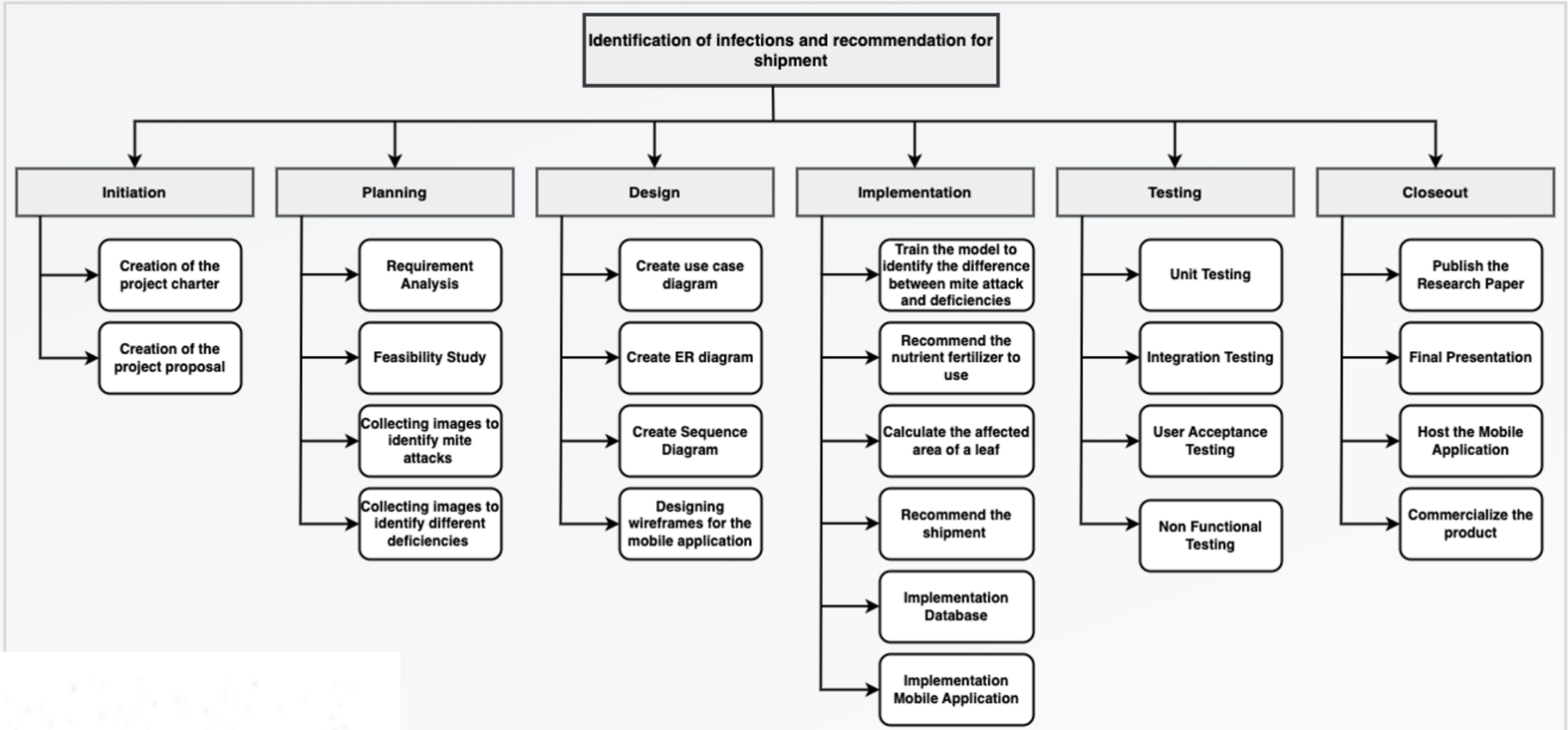
Monitoring and Decision-Making Post-Fertilization,

- Track Affliction Progress
- Assess Fertilizer Impact
- Adaptive Treatment Approach

SLIIT
FACULTY OF COMPUTING

IT19994406 | Basnayake N.S.N. | 2023-133

WORK BREAKDOWN

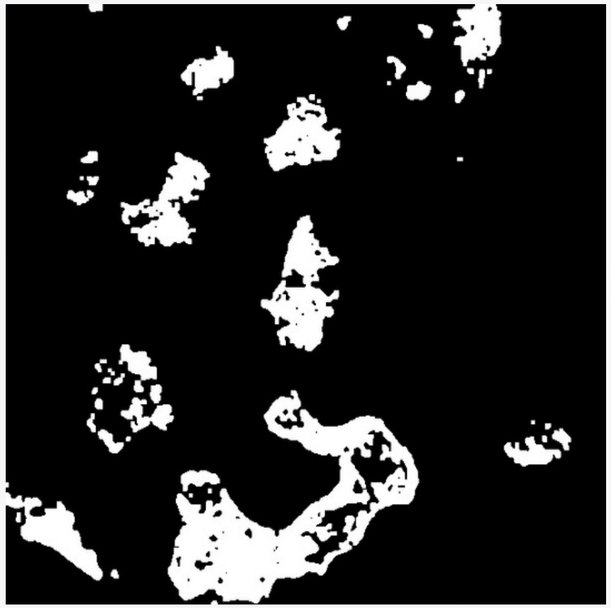


LEAF ANALYSIS:

THEN

Actual image

Mask image



SUGGESTIONS

- Remove Background
- Get accuracy

SLIIT
FACULTY OF COMPUTING

IT19994406 | Basnayake N.S.N. | 2023-133

ITERATIVE TESTING

COLOR DETECTION

Actual image

Using HSV color spaces detect brown-colored areas

Mask image

SLIIT FACULTY OF COMPUTING

IT19994406 | Basnayake N.S.N. | 2023-133

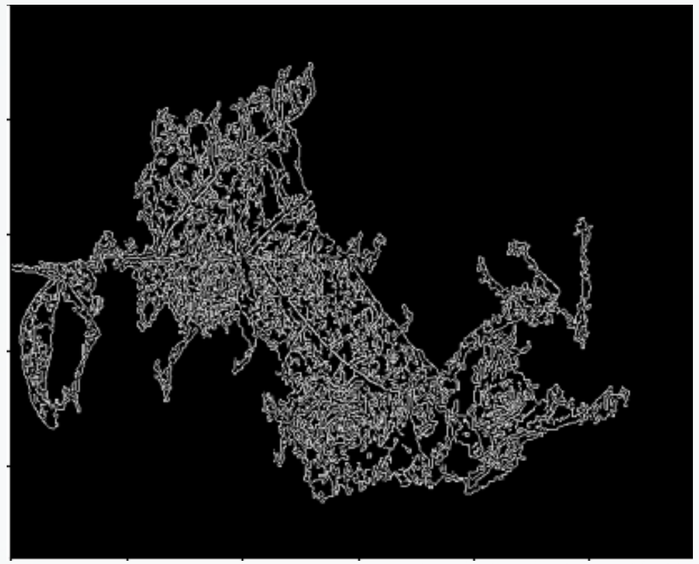
ITERATIVE TESTING

EDGE DETECTION

Actual image

By employing the Canny edge detection method detect edges

Mask image



SLIIT FACULTY OF COMPUTING

IT19994406 | Basnayake N.S.N. | 2023-133

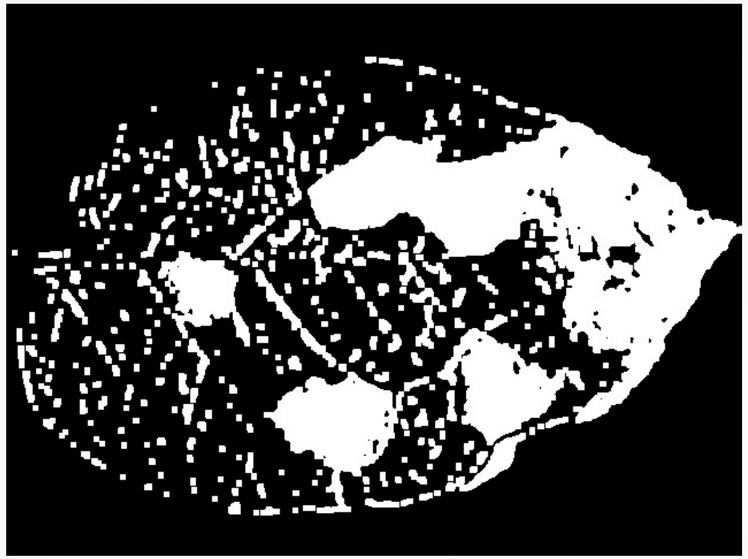
ITERATIVE TESTING

HYBRID DETECTION

Actual image

Synergize both color detection and Canny edge detection methods

Mask image



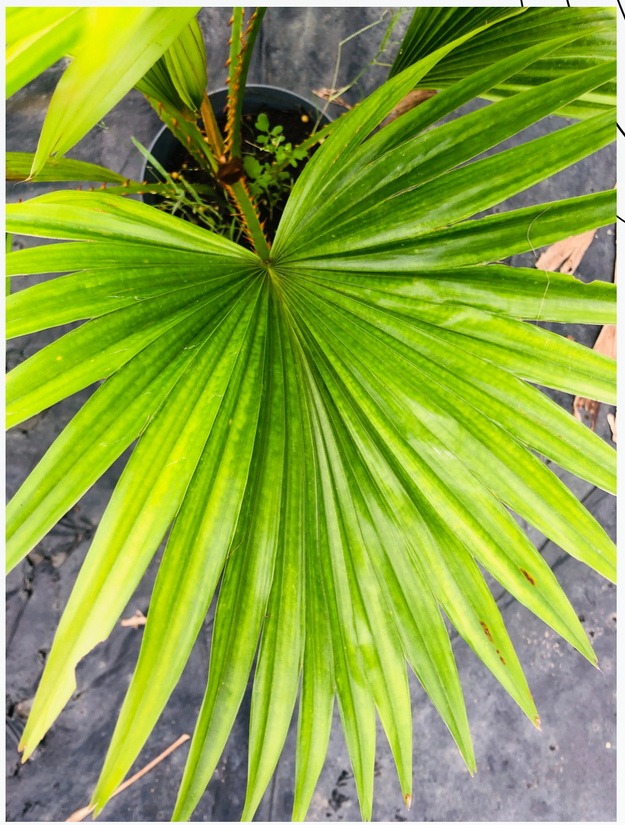
SLIIT FACULTY OF COMPUTING

IT19994406 | Basnayake N.S.N. | 2023-133

FIND THE HSV RANGE OF GREEN

Manually select regions of interest (ROIs) on the leaf image

TEST IMAGES



SLIIT FACULTY OF COMPUTING

IT19994406 | Basnayake N.S.N. | 2023-133

FIND THE HSV RANGE OF GREEN

Image name	H range		S range		V range	
	Min	Max	Min	Max	Min	Max
test1	28	39	81	255	121	255
test2	38	61	53	250	48	178
test3	34	55	63	200	101	227
test4	55	87	109	248	26	196
test5	37	76	29	255	73	255
test6	46	95	11	203	58	255
test7	39	44	83	255	67	197
test8	28	52	65	255	55	236
test9	44	73	38	239	48	224
test10	30	93	0	255	40	255
test11	29	80	4	255	96	255
test12	36	62	40	255	108	250
test13	42	91	19	255	60	255
test14	32	46	34	255	176	250
test15	37	49	60	255	112	249

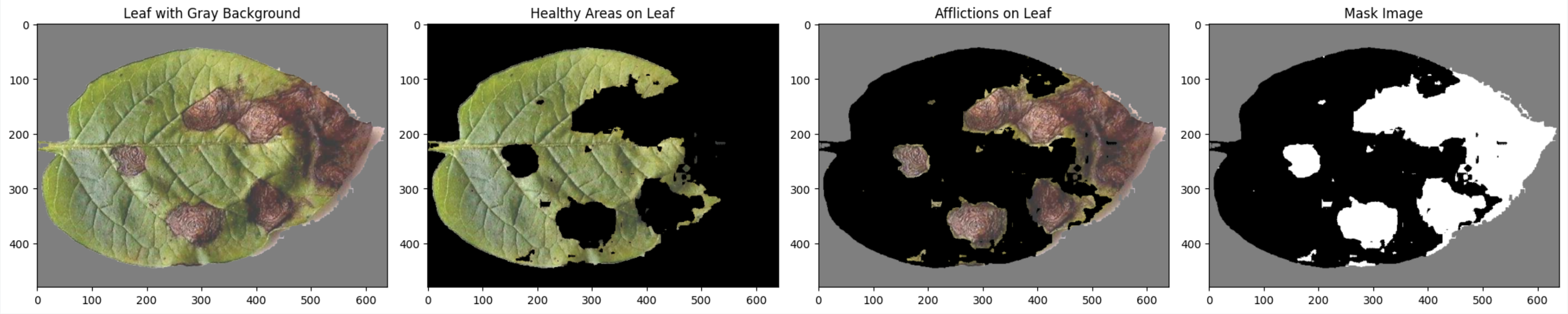
Finally getting Minimum & Maximum Values:

H : [28,95]
S : [0,255]
V: [26,255]

SLIIT FACULTY OF COMPUTING

IT19994406 | Basnayake N.S.N. | 2023-133

LEAF ANALYSIS: NOW



01) Background Removal :

- **Adaptive Thresholding and Contour Extraction**

02) Green Color Detection :

- **HSV Color Space Transformation**

03) Inverse Mask Creation :

- **Bitwise NOT Operation on a healthy green mask**

04) Final Mask Generation :

- **Bitwise Operations to combine and visualize healthy and afflicted zones**

SLIIT FACULTY OF COMPUTING

IT19994406 | Basnayake N.S.N. | 2023-133

AFFLICTION AREA QUANTIFICATION

FORMULA:

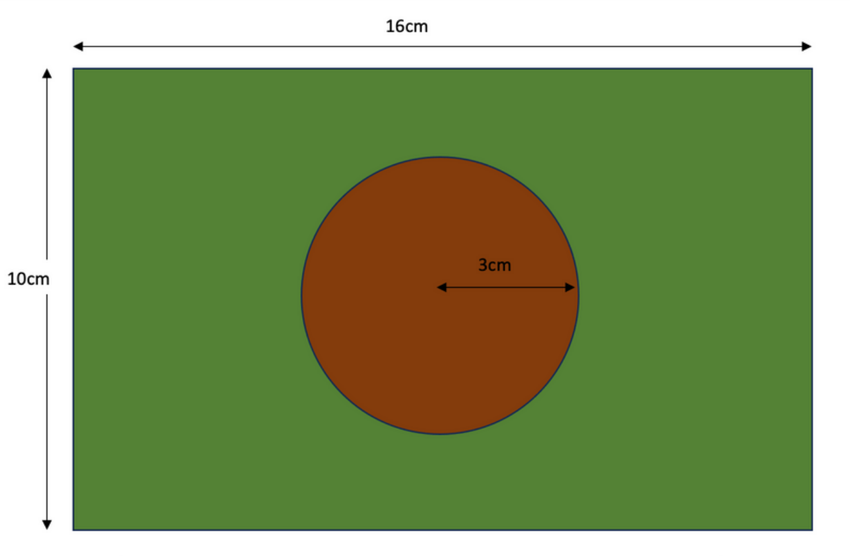
$$(\text{Afflicted Area} / \text{Total Leaf Area}) * 100$$

SLIIT
FACULTY OF COMPUTING

IT19994406 | Basnayake N.S.N. | 2023-133

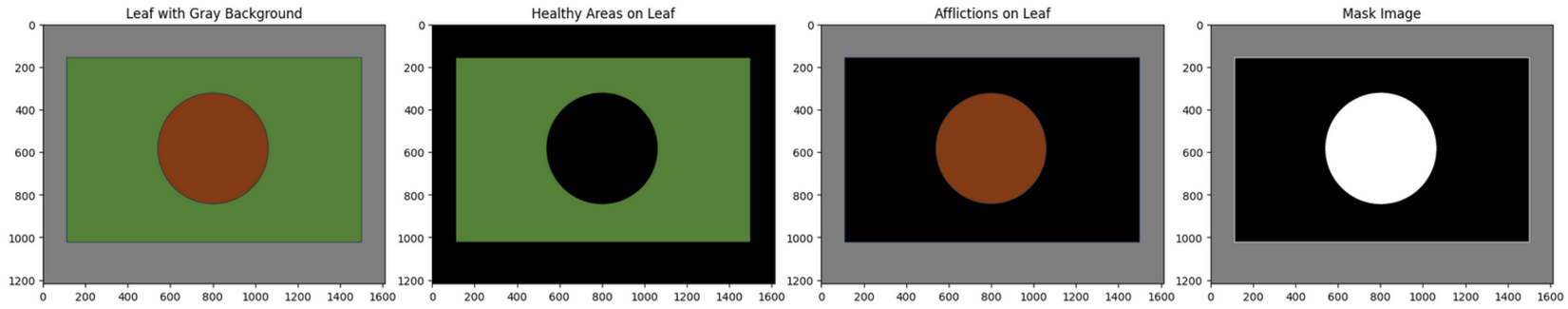
ACCURACY USING GEOMETRIC SHAPES WITH KNOWN AREAS

Actual Area



(28.29/160)x100%
= 17.68%

Computed Area



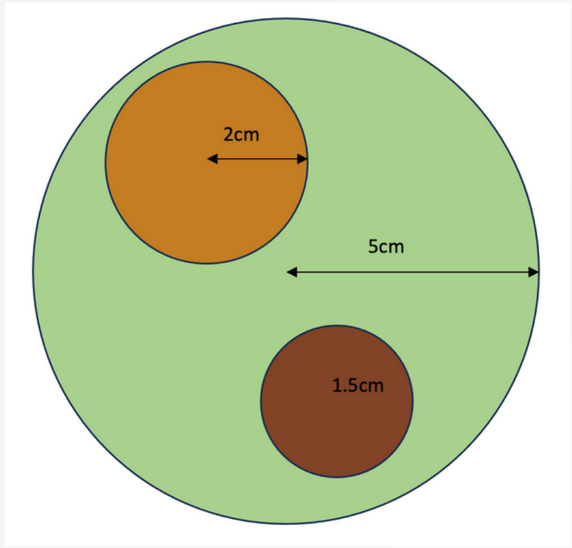
18.90%

SLIIT
FACULTY OF COMPUTING

IT19994406 | Basnayake N.S.N. | 2023-133

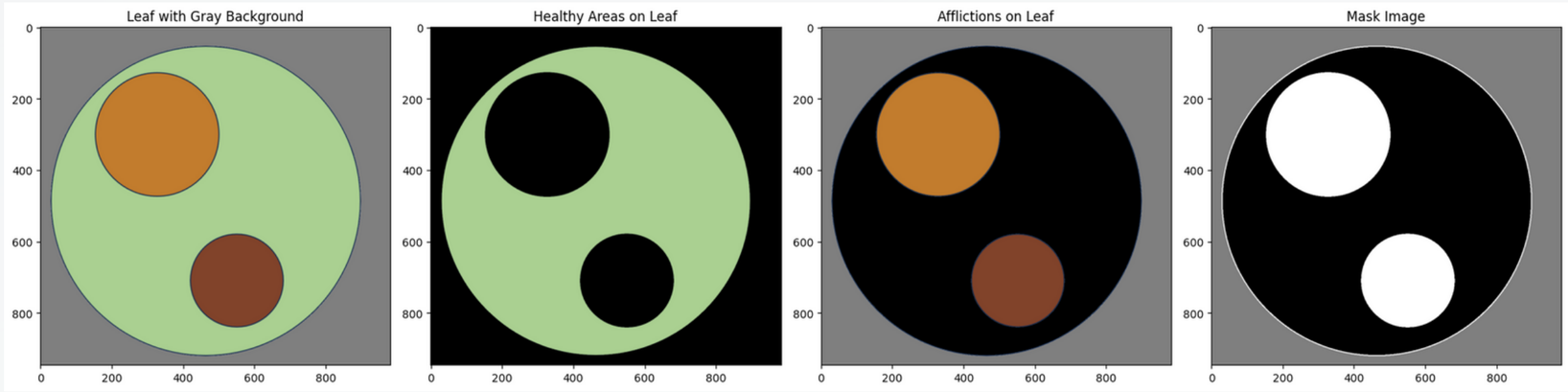
ACCURACY USING GEOMETRIC SHAPES WITH KNOWN AREAS

Actual Area



(19.64/78.57)x100% = 25%

Computed Area



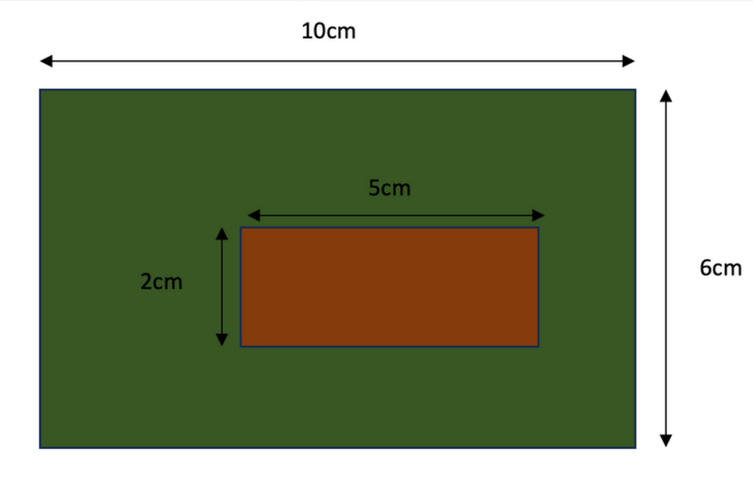
26.60%

SLIIT FACULTY OF COMPUTING

IT19994406 | Basnayake N.S.N. | 2023-133

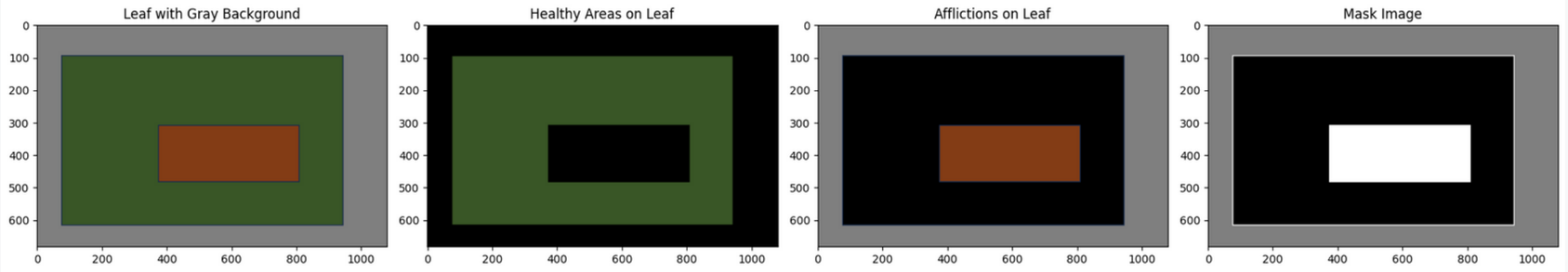
ACCURACY USING GEOMETRIC SHAPES WITH KNOWN AREAS

Actual Area



(10/60)x100%
= 16.66%

Computed Area

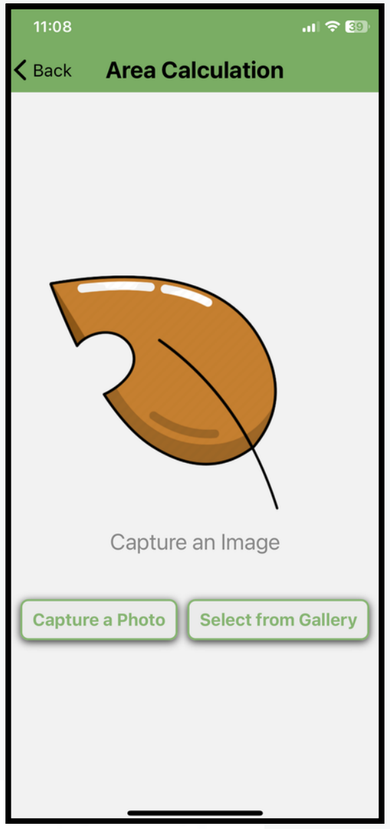


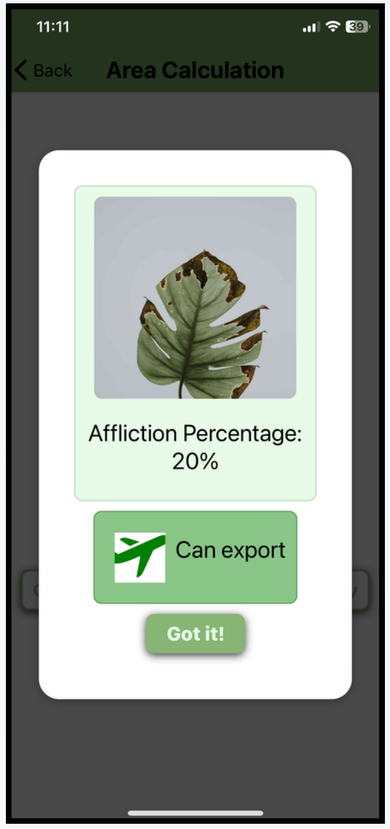
18.71%

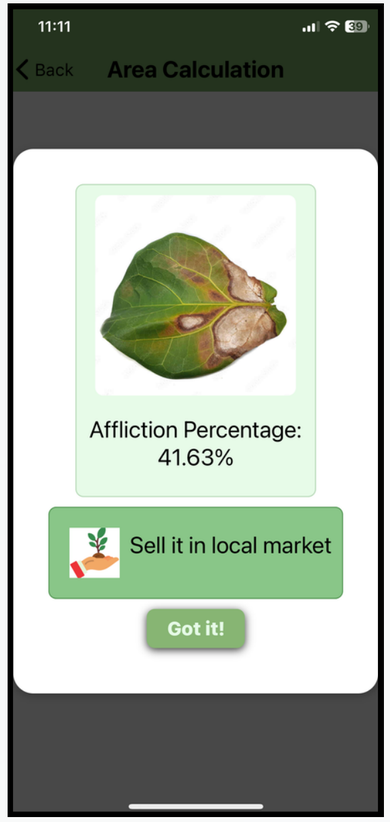
SLIIT FACULTY OF COMPUTING

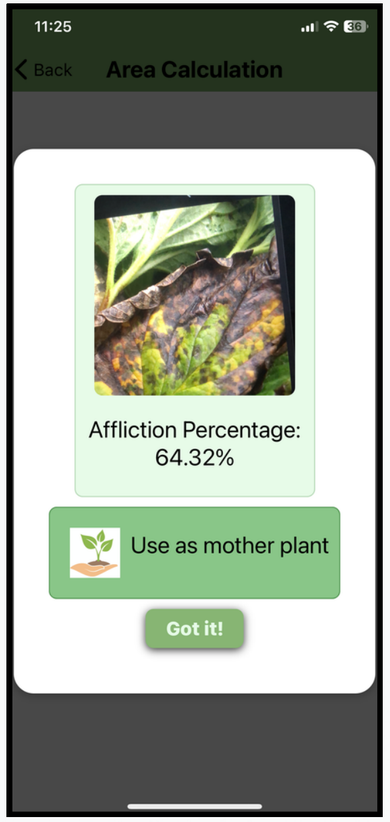
IT19994406 | Basnayake N.S.N. | 2023-133

UI FOR MOBILE APP









SLIIT
FACULTY OF COMPUTING

IT19994406 | Basnayake N.S.N. | 2023-133

NON FUNCTIONAL REQUIREMENTS

Usability

User Interface:

Allowing users to easily capture or upload images and view results

Reliability

Error Handling:

Handle potential issues, such as failed image uploads

Portability

Effortlessly manage the mobile app from anywhere.

FUTURE WORKS

1. **UI Enhancement**:

We aim to refine and further optimize the user interface, ensuring an even more intuitive and user-friendly experience across all devices.

2. Boosting Model Accuracy:

Continuous research and data collection will be undertaken to further enhance the precision of our leaf affliction detection model, aiming for near-perfect accuracy.

COMMERCIALIZATION & BUSINESS PLAN

Commodity Version

- Prediction of the current state of growth in ornamental plants
- Identification of the relative growth rate of plant height
- Distinguishing nutrient deficiencies from mite attacks
- Identification of the most suitable floriculture plant to grow based on weather and resource factors
- Forecastination of the demand for local and international markets

Premium Version (100$annually)

- Prediction of the time it will take for a plant to reach the required growth size for an order
- Providing suggestions to promote growth and advance the ornamental plant to the next level
- Identifying various floriculture crop varieties
- Recommendation of plants for shipments
- Predicting the optimal supply strategy for meeting the required output quantity based on geographical areas and finding the nearest vendor
- Providing packaging recommendations based on the plant's lifespan

COMMERCIALIZATION & BUSINESS PLAN

Target Audience:

- Floriculture businesses (small, medium, & large scale)
- Farmers and gardeners who grow ornamental plants
- Horticulture enthusiasts
- Research and educational institutions

Marketplace:

- Mobile application for Android & iOS devices
- Secure & user-friendly interface for customers & vendors

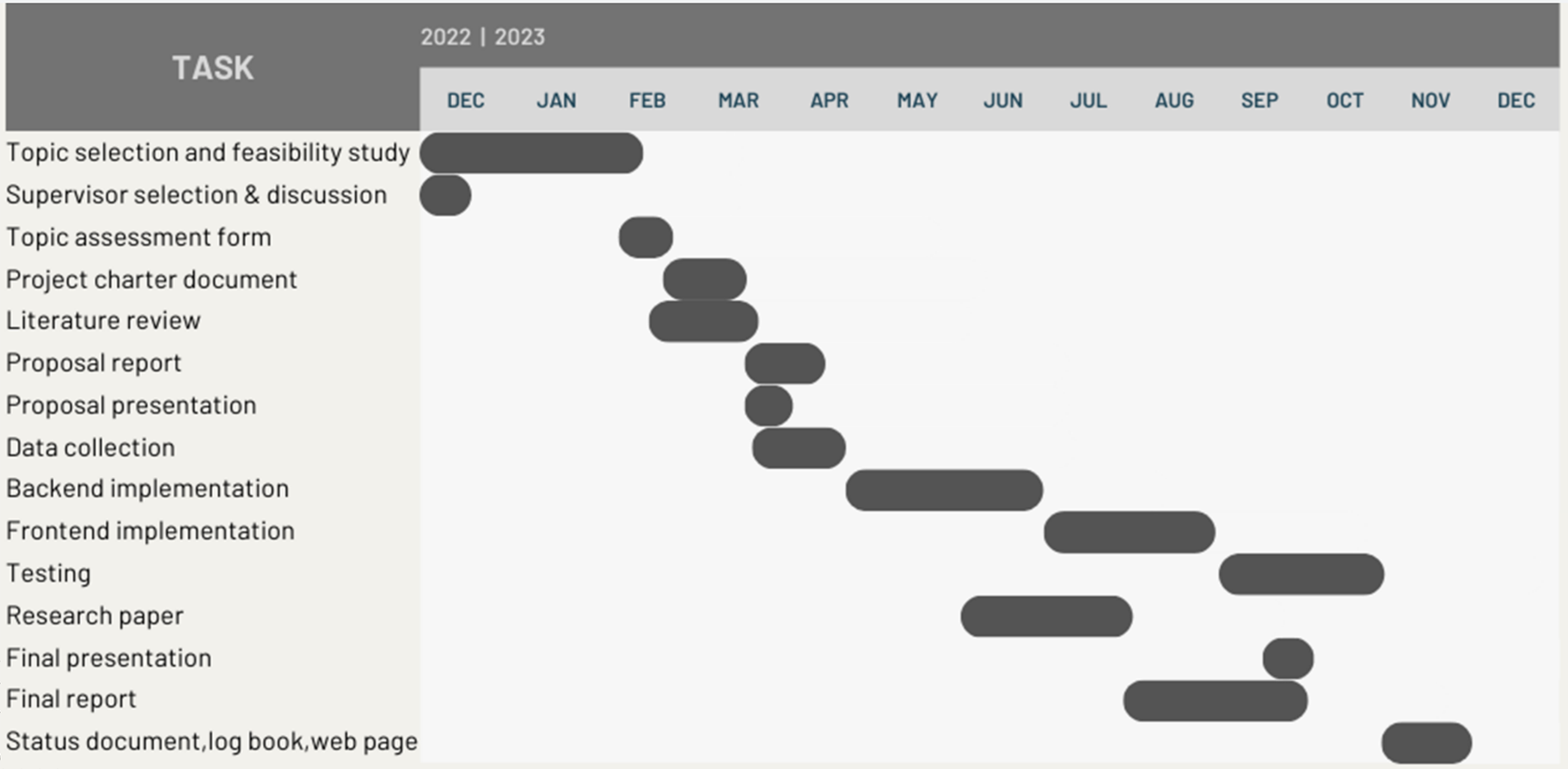
Marketing Plan:

- Social media marketing
- (Facebook, Instagram, Twitter)
- Email marketing campaigns
- Share leaflets

SLIIT
FACULTY OF COMPUTING

2023-133

GANTT CHART



SLIIT
FACULTY OF COMPUTING

2023-133

THANK YOU

PlantPal