AGRIPRENEURS – MOTIVATING INDIVIDUALS TO BECOME ENTREPRENEURS THROUGH AGRICULTURE

Project ID – 2021-090

Ampitiya Gedara Janitha Lahiru Pramodh Rajapakse

(IT17023610)

Bachelor of Science (Hons) in Information Technology

Specializing in Software Engineering

Department of Software Engineering

Sri Lanka Institute of Information Technology
Sri Lanka

August 2021

AGRIPRENEURS – MOTIVATING INDIVIDUALS TO BECOME ENTREPRENEURS THROUGH AGRICULTURE

Ampitiya Gedara Janitha Lahiru Pramodh Rajapakse

(IT17023610)

Dissertation submitted in partial fulfillment of the requirements for the Bachelor of Science (honors) degree in IT (Specializing in Software Engineering)

Department of Software Engineering

Sri Lanka Institute of Information Technology
Sri Lanka

August 2021

Declaration

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. Also, I hereby grant to Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute my dissertation in whole or part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as article or books).

Signature: ……. Date: ……

Signature of the supervisor: ……. Date: ……

Abstract

Modern technology is being used to build a bridge between the agriculture and entrepreneurship domains. This is accomplished by incorporating a motivational gamification paradigm into the system, which allows users to interact not only with the application but also with other users. An Internet of Things module that analyzes soil data such as humidity and temperature. A machine learning model is used to assess user behavior in this integrated auction platform. An image processing approach that can help users in uncontrollable conditions like plant diseases. The purpose of this study is to enable working or semi-employed people to earn a passive income.

The aforementioned component will help users gain perspective on which crops to invest in by seeing how the crop of the price fluctuates. Furthermore, the user can engage directly with buyers and sellers in order to meet their needs for agriculture.

Acknowledgment

I would like to express my appreciation and heart felt gratitude to our supervisor Mrs. Dhammika Perera and co-supervisor Ms. Janani Thamalaseelan who helped us to succeed in this research and for sharing their own experiences and expertise with the project matters. We extend our gratitude to the Sri Lanka Institute of Information Technology (SLIIT) for giving an opportunity to carry out this research project. A special thanks to the Agriculture department for providing datasets and our friends and families, who helped us by giving suggestions for our research. We would also like to extend our gratitude to the members of the panel for sharing their ideas and helping us be on the right path throughout the development of this project.

Table of Contents

List of Tables

List of Figures

1. INTRODUCTION

Agricultural entrepreneurship is a new field that is gaining popularity. It entails examining and comprehending agricultural entrepreneurs' strategies, particularly in response to institutional changes as well as economic and technical disruptions in the agricultural industry. Despite the growing number and diversity of publications, there is still a need for a deeper understanding of agricultural entrepreneurship dynamics from the perspective of entrepreneurial theory. Although this subject is quite popular among developed countries such as Australia, developing countries such as ours lack knowledge on this topic.

If an individual is properly educated on the subject of agricultural entrepreneurship and is provided with the right tools, with the right learning curve he or she can excel greatly and rapidly as an entrepreneur. While agricultural entrepreneurship helps an individual prosper, it also elevates the economic growth of a country. Sri Lanka has for centuries been a country that was self-sustained due to its rich soil, although recently many individuals tied to the agricultural industry has been moving away from it rather than towards it.

The main reason for this move away from the agricultural sector is due to the prices fluctuating of vegetables and fruits. Suppliers complain that they do not get a fair price for the time and effort they invest in their crops. Furthermore, access to vegetables during this pandemic has also been a challenge as many suppliers could not transport their crops. Bridging the agriculture and entrepreneurial sectors in developing countries will not only help to increase the economy of the country, but it will also allow unemployed people in countries like Sri Lanka to be semi-employed and benefit from the system. Due to the failing economy in developing countries such as Sri Lanka, farmers and other industries are being compelled to plant vegetables, fruits, and other crops in order to boost the country's export output, as well as boost the economy and feed the country's population. According to other research conducted the GDP for agriculture has be gradually dropping throughout the past decade from 17.2% in 2005 and 11.1% during 2012, and has been dropping even further [1].

Therefore, we offer our project as a remedy to the foregoing problems in traditional agriculture industry as well as a tool for the elevation of the Sri Lanka's economy. The main research problem is to assist and encourage individuals to engage in the agricultural domain by showing the rewards of this sector. Once an individual sees profit and potential in the project they would be invested and continue the project to become a successful entrepreneur.

The tool we develop will be a mobile application that will be composed of 4 major functionalities namely, the motivational gamification model, IoT soil analysis model, robust image processing model and also a prediction algorithm paired with an integrated auction platform.

1.1. Background Literature

Within the recent years, multiple systems have been presented in several countries to implement crop price prediction using machine learning techniques, and they have used various machine learning approaches for prediction. The below mentioned projects have been thoroughly analyzed and in order to get a better understanding of the literature of this subject.

1.1.1. Machine Learning for Price Prediction for Agricultural Products

In a review conducted, various machine learning algorithms were reviewed in order to compared the rate of accuracy of these models when used to predict prices for agricultural products. They aimed to determine the best algorithm that can be used for a scenario such as ours. The machine learning models used for price prediction vary between regions and agricultural products but also vary due to the availability and diversity of the data. Thus, establishing the real value of any model is nearly impossible. Even though Neural Networks showed a higher level of success during the evaluation period, it cannot be concluded that it is the best prediction model in all cases [2].

1.1.2. Crop price prediction system using machine learning algorithms

Pandit Samuel et al… proposed the above system which will predict the price of a certain crop using data from the World Supply and Demand Estimate (WSDE), which is where the data for the research comes from. After that, the data is cleaned and transformed through a series of stages before being prepared for analysis. Data Analytics techniques were adopted in order to estimate crop price analysis with existing data. Linear Regression is employed for discovering important information from the agricultural datasets. Neural Network is constructed for price prediction to increase the accuracy percentage. The root mean square error is calculated for every technique to accurately measure the accuracy of each system employed and the most accurate system is then selected [3].

1.1.3. Predicting yield of the crop using machine learning algorithm

Ranjani Dhanapal et al… mentions that various algorithms can be used for crop price prediction such as decision trees, support vector machines, neural networks, deep learning, etc. Our model

used a supervised machine learning algorithm called Decision tree Regressor. It is trained on several Kharif and Ragi crops (Paddy, Wheat, Cotton, Barley,etc.) providing better accuracy. The Model further can be trained with climate-aware farming techniques, provide fertilizer suggestions, and identifying systems of crop monitoring, warning on pest outbreak, disease outbreak based on advanced AI Models [4].

1.2. Research Gap

While many of the above-mentioned proposed systems seem to be promising, none of these systems have made available a tool while uses both accurate crop price prediction and an auction platform. These two functionalities will both help the entrepreneur in making decisions on what

crops to invest in and also motivate them to engage in agricultural entrepreneurship with credible data on crop prices.

Tools such as the FieldCheck App and the Descrates Crop App are among the most popular when considering tools that help agricultural entrepreneurs. The below mentioned table compares both the FieldCheck App and Descrates Crop App with our proposed solution Agripreneurs.

Features	**FieldCheck App**	**Descartes Crop App**	**Agriprenuers**
Crop visualization	Yes	No	Yes
Yield Forecast	No	Y	Y
Trending Crops	No	No	Yes
Target Audience	Farmers	Farmers	Anybody/Entrepreneurs
Suitable Crop recommendation	No	No	Yes
Disease detection and solutions	No	No	Yes
Auction Platform	No	No	Yes
IoT analysis model	No	No	Yes
Motivational Model	-	-	Yes

Table 1: Comparison of existing applications with agriprenerus application

1.3. Research Problems

The main research problem that we recognized was the problem of how an individual should be motivated into engaging him or herself in an agricultural entrepreneurship project by the means of our tool provided. Although many individuals already are equipped with the resources required in order to engage in agriculture in their own backyards the lack of drive results in none of them even giving it a thought. The figure below illustrates the availability of bare land in the city of Malabe in Sri Lanka.

Figure 1: Bare land availability in Malabe

To overcome the above problem, we developed a mobile application equipped with state-of-the-art technologies that can help anyone shape themselves into an agricultural entrepreneur. We aim to integrate functionalities that would keep the user engaged in this application and achieve the maximum output for their efforts.

1.4. RESEARCH OBJECTIVES

The major purpose of this project is to establish a platform that will help the country generate more entrepreneurs by bridging the gap between agriculture and entrepreneurship through technology. Each and every functionality of the application will be highly dependent on one another as the seamless flow of data is required in order to keep the user engaged and motivated to become an entrepreneur.

1.4.1. Main Objectives

The two main objectives in this specific research component is to build a machine learning model that can accurately predict the crop prices based on previous data and also build an integrated auction platform that will enable users to create auctions and make bids to already available auctions for crops. These two features will play a major role in being motivational factors that will ultimately contribute to our main goal which is to help motivate the user to become an agricultural entrepreneur.

1.4.2. Specific Objectives

- Prediction Model

  - Reviewing available machine learning algorithms in order to decide on the best algorithm for crop price prediction.
  - Collecting appropriate data for the training of the machine learning model.
  - Cleaning collected data in order to be fed into the machine learning model.
  - Building the machine learning model using modern practices while elevating accuracy.
  - Testing of the machine learning model

- Increasing accuracy of the machine learning model using machine learning practices learnt during the machine learning module
- Deploying the trained model so that it is accessible through cloud services
- Integrating the machine learning model with the mobile application in order to get predictions.

- Auction Platform

- Designing appropriate architecture for the application using software engineering practices.
- Designing user friendly interfaces for the auction platform.
- Building mobile application with auction platform integrated by referring the designed user interfaces.
- Building database model for the auction data to be stored.
- Building backend project for the seamless flow of data and real time updates of auctions.
- Testing the flow of data throughout the auction platform.

2. METHODOLOGY

2.1. Component Overview

The mobile application which is our ultimate proposed solution consists of four major areas which are namely the gamification model developed using psychological theories, an IoT data analysis model where data will be collected and analyzed through a device which will give accurate analysis to the user, a robust image processing model that will predict diseases of crops the users have and propose remedies for these diseases and finally an auction platform that will also contain the functionality of predicting future trending crops to the user.

The component that I will be developing is the integrated auction platform with the functionality of predicting future trending crops.

2.2 Methodology

The below figure illustrates the general architecture of our application.

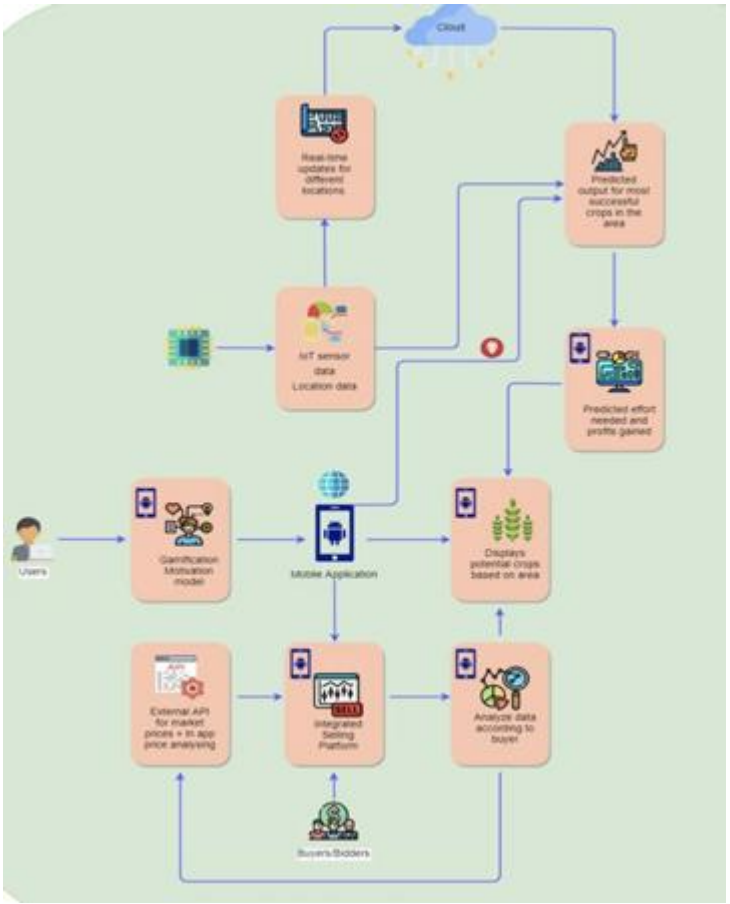


Figure 2: High Level System Diagram

According to the above illustrated component's flow, there are just a few actions that must be completed in order to construct a successful model.

- Data cleaning and preprocessing, investigating the data and handling missing values.
- Conducting a thorough Time Series Analysis
- Using Prophet to forecast sales for the upcoming weeks.

2.1.1. Resources Needed

2.1.1.1 Software Boundaries

1. Visual Studio Code

Visual Studio Code is a lightweight yet capable source code editor for Windows, macOS, and Linux that runs on your desktop. It contains built-in support for JavaScript, TypeScript, and Node.js, as well as a large ecosystem of extensions for additional languages and runtimes (such as C++, C#, Java, Python, PHP, and Go) (such as .NET and Unity).

2. Jupyter Notebook

Jupyter Notebook is an open-source web software that lets you create and share documents with live code, equations, visualizations, and narrative text. Data cleansing and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and many other applications are all possible.

3. MongoDB

The most powerful cloud database service available, with unrivaled data distribution and mobility across AWS, Azure, and Google Cloud, as well as built-in automation for resource and workload optimization.

4. Anaconda Environment

Anaconda is a Python and R programming language distribution for scientific computing that seeks to make package management and deployment easier.

5. Kaggle

This is a best site that contains many datasets that are publicly available which we can use in order to train our machine learning models.

2.1.1.2 Hardware Boundaries

- A machine with a high processing power required for the hosting of the server.
- A mobile phone for testing the auction platform.
- The NodeMCU model that will be built to detect the soil and air conditions should be purchased to have more benefits from the application.

2.1.1.3 Communication Boundaries

An internet connectivity will be essential for this application to function since the databases and backends will be hosted therefore API requests will have to be send through the network in order to access the required service. Locale provided for this application will be English.

2.1.1.4 Memory Constraints

Due to the training of machine learning models this application will require a lot of memory space initially during the training phase of these models. Once trained these models can then be deployed and services can be accessed through a cloud service.

Back-end

- RAM - 8 GB minimum, 16 GB or higher is recommended.
- Data Storage – Minimum 4GB

Front-end

- RAM - Minimum of 4GB
- Data Storage – 400 Mb or higher

2.1.1.5 Operations

The user is able to follow the aforementioned operations.

- Add auctions: user can add auctions for their own crops by adding data such as a starting price and images for the crop.
- View ongoing auctions: user can view the ongoing auctions for crops that other users have posted.
- Bid on ongoing auctions: users can bid on ongoing auctions by adding a bid price
- Delete auction: users can delete an auction that they have posted.
- View predicted crop price: users can view the predicted prices for crops

2.1.1.6 Site adaptation requirements

We plan to release the application with the English locale initially. Local languages such as Tamil and Sinhala will be available as improvements in future versions. Furthermore, the mobile should be connected to the internet in order to acquire access to the database and services of the deployed backend server.

Agripreneurs is a mobile application and this can be easily installed on any device therefore no other external dependencies exist for the functionality of this project.

2.1.1.7 User Interfaces for the auction platform

Figure 3: Main screen for auction platform

Figure 4: Auction Detail Screen

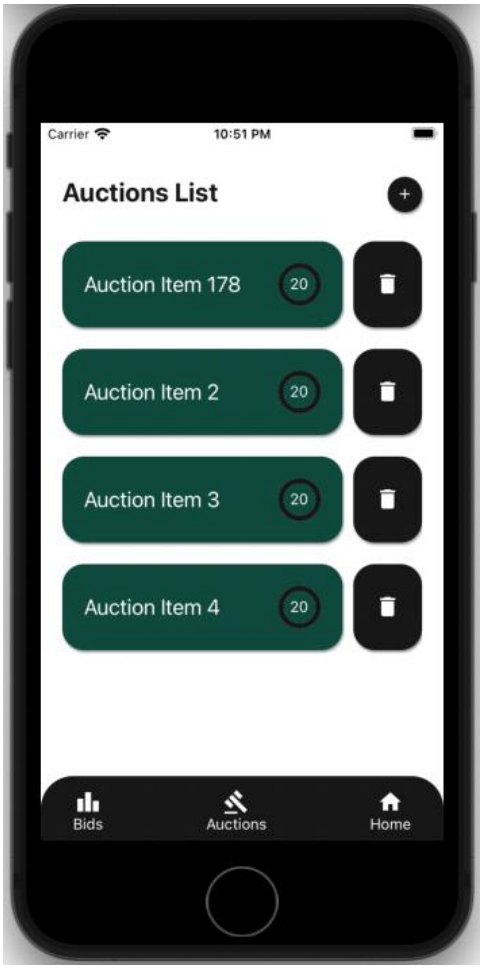


Figure 5: User's Auction List Screen

Figure 6: User Bids Screen

2.2. Commercialization

2.2.1. Basic User

On its initial release, Agripreneurs is a free program. The app, on the other hand, has its own integrated buyer-seller platform. Individual users of the application can access this feature to a certain extent. The free users have access to all of the application's components, including the IoT data analysis model, which must be purchased individually by the users. All users will have full access to the gamification concept, image processing model, and limited auction platform.

2.2.2. Premium User

As an added feature, premium users of the app will be able to see data visualized within the app. Users can also export their reports and evaluate the information as needed. Users can up grade to the premium edition to make better decisions about prices and crops to cultivate based on the application's historical data. In addition, the application will take a cut of all transactions made through the buyer seller platform. If the user upgrades to the premium version, the percentage of the cut that will be taken will be lowered. The first percentage cut will be 10%, but after becoming a premium user, it will be decreased to 7.5 percent.

2.2.3. Advertisements

An integrated advertisement platform is also created to show advertisements using Google AdSense, where the developers of the application can simply earn an extra penny. Other agricultural products and equipment, if needed, can also be auctioned through the integrated buyer-seller auction platform.

Only the agriculture domain is being evaluated in the initial stages of the application. The technique, however, can be used to a wide range of areas, including textiles, floral arrangements, and decorations. Handcrafted objects of any kind can be included in the platform. In addition, dry foods like rice and gram food products can be added to the system. As a result, the country's farmers may benefit fully from the system.

2.3. Testing & Implementation

2.3.1 Testing

Software testing is crucial in the development lifecycle since it identifies problems and errors that occurred throughout the development process. It's also critical to ensure that the product is of high quality. Testing is necessary for a software application's or product's effective performance. Each aspect of the software development life cycle, such as unit testing, integration testing, system testing, and user acceptability testing, requires testing.

For the individual component developed, testing was carried out for 3 systems;

- Expo application (Frontend)
- Flask Server (Backend)

Frontend Testing and Backend Testing: Following are some test cases conducted to make sure the mobile application is performing well without any failures that could result in the app crashing.

Test case ID	01
Test case scenario	Add auction
Test steps	a. User enters auction details b. User submits auction
Test Data	Auction Item Name – Tomatoes Starting Price – 100.00 Expiry Date – 20/11/2021 Images – Sample Image List
Expected Results	The auction should be successfully submitted and displayed on the auction list.
Actual Results	Auction is submitted and displayed to auction list.
Pass/ Fail	Pass

Table 2: System Test case1

Test case ID	02
Test case scenario	Add auction with invalid details
Test steps	a. User enters auction details b. User enters invalid starting price, and invalid date c. User submits auction
Test Data	Auction Item Name – Tomatoes Starting Price – price Expiry Date – 20/11/2021 Images – Sample Image List
Expected Results	The user should not be able to submit the auction and should be prompted of the invalid details.
Actual Results	Same behavior as expected results.
Pass/ Fail	Pass

Table 3: System Test case2

Test case ID	03
Test case scenario	Add Bid
Test Steps	a. User selects ongoing auction b. User enters bid amount c. User submits bid
Test Data	Bid Price – 150.00
Expected Results	Bid should be shown in the auction detail page.
Actual Results	Same behavior as expected results.
Pass/ Fail	Pass

Table 4: System Test case3

Test case ID	04
Test case scenario	Add Bid with invalid details
Test steps	a. User selects ongoing auction b. User enters bid amount as string c. User submits bid
Test Data	Bid Price = 'price'
Expected Results	User should not be able to add bid and will be prompted about invalid input.
Actual Results	Same behavior as expected results.
Pass/ Fail	Pass

Table 5: System Test case4

Test case ID	05
Test case scenario	Delete Auction
Test steps	a. User navigates to my auctions screen b. User deletes desired auction from auction list
Test Data	
Expected Results	Auction should be deleted and removed from system.
Actual Results	Same behavior as expected results.
Pass/ Fail	Pass

Table 6: System Test case4

Test case ID	06
Test case scenario	Delete Bid
Test steps	a. User navigates to my bids screen b. User deletes desired bid from auction list
Test Data	
Expected Results	Bid should be deleted and removed from system.
Actual Results	Same behavior as expected results.
Pass/ Fail	Pass

Table 7: System Test case4

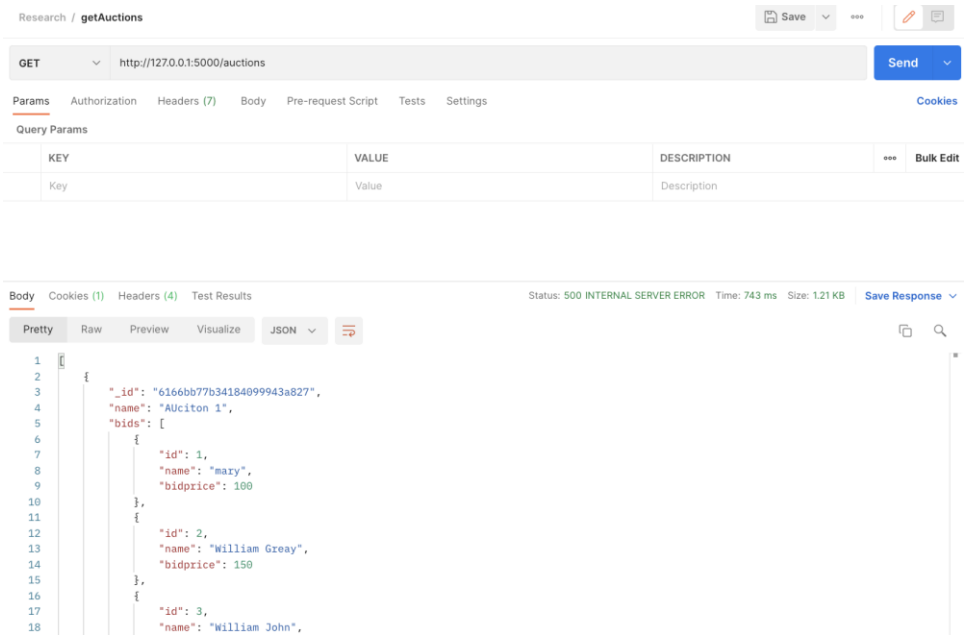


Figure 7: Get auctions postman endpoint screen

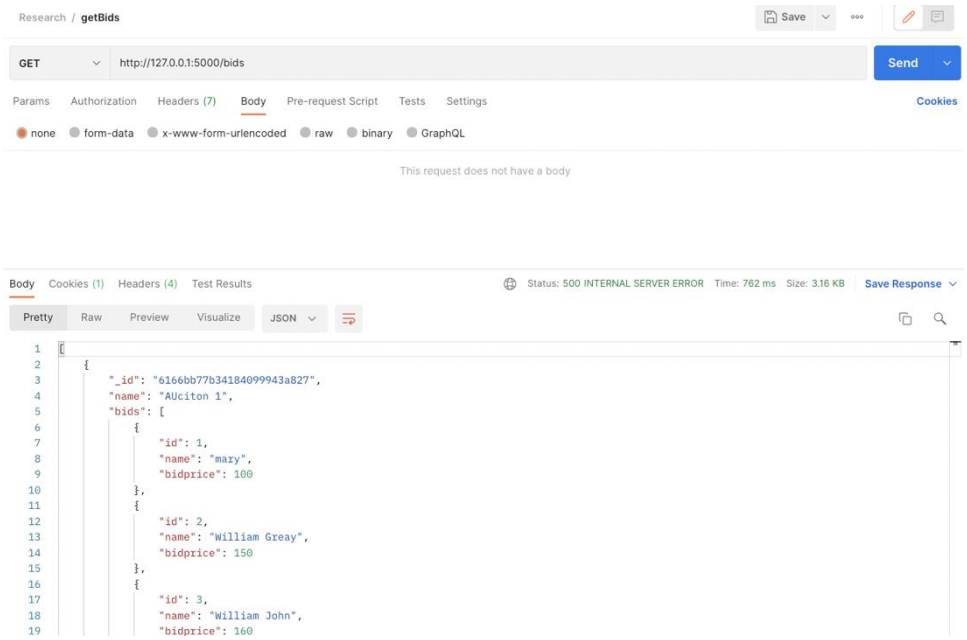


Figure 8: Get bids postman endpoint screen

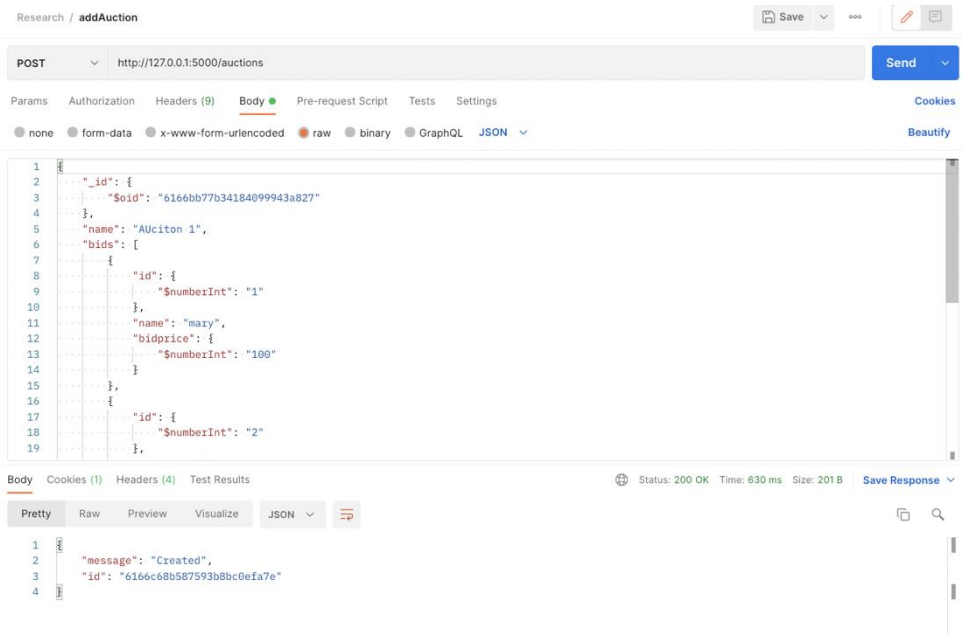


Figure 9: Add auction postman endpoint screen

2.3.2 Implementation

2.3.2.1 Data Collection

In order for the machine learning model to be trained accurately a dataset with many values was required. The algorithm used for the training of the machine learning model was the Prophet algorithm developed by Facebook. The dataset needed to contain data collected over a period of time with the target characteristic which we will be predicting. The figure below illustrates the first few rows of the dataset that I have used to train the model.

	Store	DayOfWeek	Sales	Customers	Open	Promo	StateHoliday	SchoolHoliday
Date								
2015-07-31	1	5	5263	555	1	1	0	1
2015-07-31	2	5	6064	625	1	1	0	1
2015-07-31	3	5	8314	821	1	1	0	1
2015-07-31	4	5	13995	1498	1	1	0	1
2015-07-31	5	5	4822	559	1	1	0	1

Figure 10: Dataset sample used to train the model

2.3.2.2 Data Analysis

Before actually preprocessing the data, it was essential that a good understanding of the dataset was acquired. In order to do this I used the empirical cumulative distribution function, by using this function I could obtain a clear impression of how continuous variables behave in the dataset used.

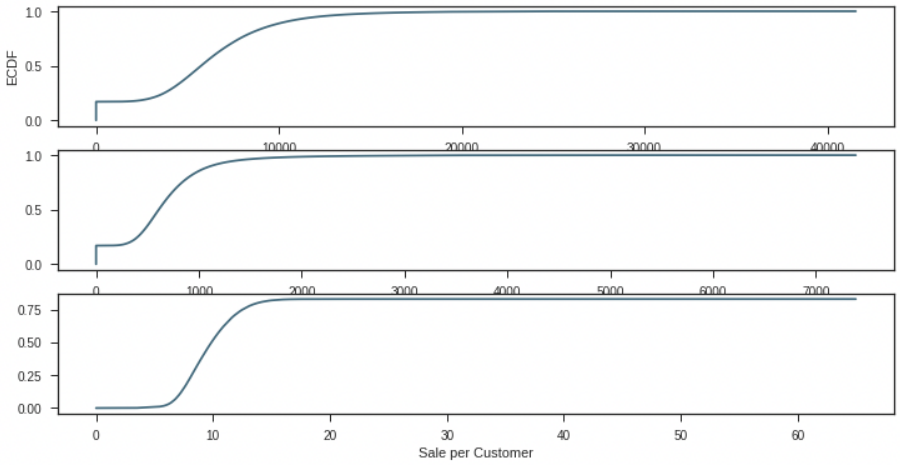


Figure 11: Plot generated from the ECDF

2.3.2.3 Data Preprocessing

The dataset obtained should always be cleaned before feeding it into the machine learning model for training. This improves the overall accuracy of the model by ensuring that we do not feed in empty values and such broken data.

3 RESULTS & DISCUSSIONS

3.1 Results

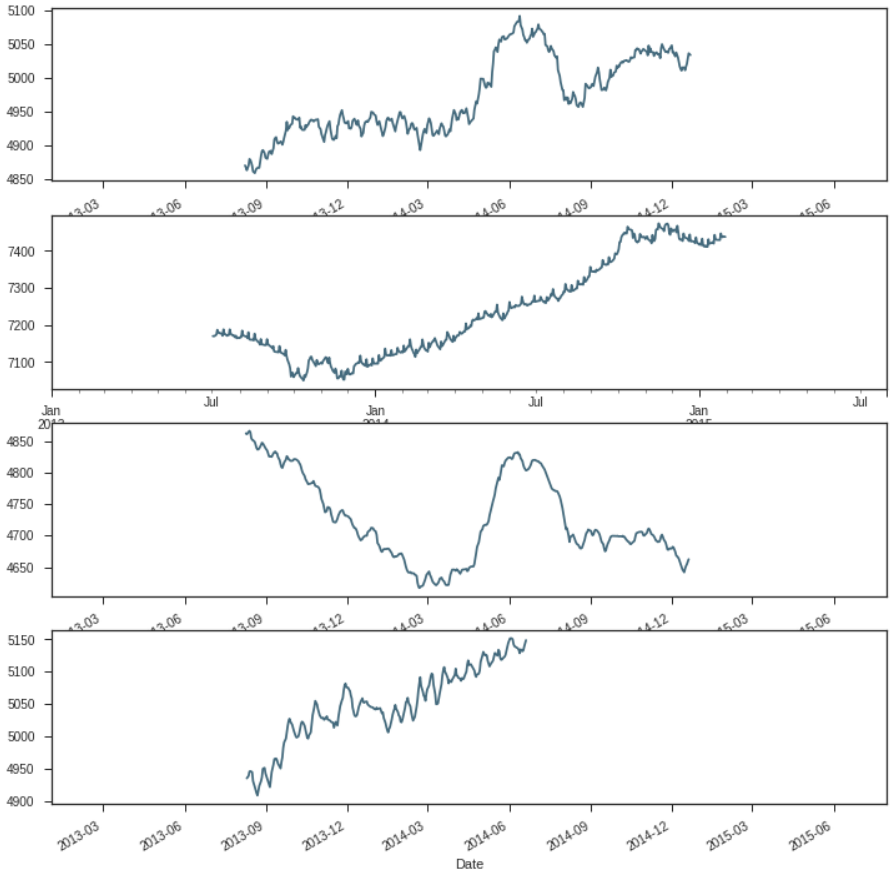


Figure 12: Overall price prediction plot

Overall prices seemed to increase after making the predictions for four types of data, however not for the third type of data (a third from the top).

3. 2 Research Findings

Crop price prediction has not been done in recent times as the amount of data available is very limited. Most crop price predictions used regression algorithms for training although it was discovered that the best possible algorithm for the prediction of prices was the Prophet algorithm developed by Facebook.

3.3 Discussion

Future Ideas

This machine learning model can be further developed in order to increase accuracy by feeding real time data from the auction platform into the machine learning model. By collecting data for how crop price varies throughout a year, accurate predictions can be made as the data that will be collected will also be viable and focused only on Sri Lanka.

4 CONCLUSION

The goal of this research study is to create a large number of self-made business people and entrepreneurs in Sri Lanka. The application will serve as a vital platform for those who wish to invest their time and effort in developing their own farm business domain while also being able to work full-time. Furthermore, this product will not only produce many self-made businesses in the agriculture industry, but it will also aid in the country's economic development by reducing the amount of imported food owing to domestic food production.

The prediction model for crop prices was a success along with the auction platform, these two functionalities while paired with the other major functionalities can prove to be a invaluable tool to any individual that has the desire to transform him/herself into a self-made agricultural entrepreneur.

Furthermore, in the event of a pandemic, such as the current position in which the country finds itself. Due to government food supply constraints, these agriculture/plantations can even help families meet their fundamental needs.

As a result, the employment of information technology in conjunction with the agricultural domain to anticipate a success rate figure will encourage people to participate in this agripreneur business domain.

REFERENCES

[1] Ft.lk. (2019). Sri Lanka heading for an agricultural issue in 2017? [Online] Available at: http://www.ft.lk/columns/sri-lanka-heading-for-an-agricultural-issue-in-2017/4-588715.

[2] Bayona-Oré, Sussy & Cerna, Rino & Hinojoza, Eduardo. (2021). Machine Learning for Price Prediction for Agricultural Products. WSEAS TRANSACTIONS ON BUSINESS AND ECONOMICS. 18. 969-977. 10.37394/23207.2021.18.92.

[3] Pandit Samuel,etal. "Crop Price Prediction System using Machine learning Algorithms." Quest Journals Journal of Software Engineering And Simulation, Vol. 06, No. 01, 2020, Pp. 14-20.

[4] Ranjani Dhanapal et al "Crop Price Prediction using supervised machine learning algorithms" 2021 J. Phys.: Conf. Ser. 1916 012042

APPENDICES

User bid screen

```
// Example of Splash, Login and Sign Up in React Native
// https://aboutreact.com/react-native-login-and-signup/

// Import React and Component
import React from 'react';
import {View, StyleSheet, Text, ScrollView} from 'react-native';
import { AppAuctionItem } from '../components/auction_item.component';
import { AppContainer } from '../container/container';
import themeColors from '../assets/colors';
import { AuctionBottomTab } from '../Navigations/AuctionBottomTab';

const UserBidScreen = ({navigation}) => {
  return (
    <View style={styles.container}>
      <AppContainer>
        <Text style={styles.auctionHeading}>My Bids</Text>      You, seconds ago • Uncommitted changes
        <ScrollView>
          <View>
            <AppAuctionItem label='Auction Item 1' onPress = {() => navigation.navigate("SignUp")}/>
            <AppAuctionItem label='Auction Item 2' />
            <AppAuctionItem label='Auction Item 3' />
            <AppAuctionItem label='Auction Item 4' />
          </View>
        </ScrollView>
      </AppContainer>
      <AuctionBottomTab/>
    </View>
  );
};
export default UserBidScreen;

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#ffffff'
  },
  header: {
    height: '35%',
    marginBottom: 25
```

User auction list screen

```
import React from 'react';
import {View, StyleSheet, Text, ScrollView, TouchableOpacity} from 'react-native';
import { AppAuctionItem } from '../components/auction_item.component';
import { AppContainer } from '../container/container';
import themeColors from '../assets/colors';
import { AuctionBottomTab } from '../Navigations/AuctionBottomTab';
import { MaterialIcons } from '@expo/vector-icons';

const UserAuctionListScreen = ({navigation}) => {
  return (
    <View style={styles.container}>
      <AppContainer>
        <View style={styles.header}>
          <Text style={styles.auctionHeading}>Auctions List</Text>
          <TouchableOpacity onPress={() => navigation.navigate("AuctionCreateScreen")} style={styles.addBtn}>
            <MaterialIcons name="add" size={16} color={themeColors.WHITE} />
          </TouchableOpacity>
        </View>
        <ScrollView>
          <AppAuctionItem label='Auction Item 178' delete onSelect = {() => navigation.navigate("AuctionDetailScreen")}/>      You, 2 days
          <AppAuctionItem label='Auction Item 2' delete/>
          <AppAuctionItem label='Auction Item 3' delete/>
          <AppAuctionItem label='Auction Item 4' delete/>
        </ScrollView>
      </AppContainer>
      <AuctionBottomTab/>
    </View>
  );
};
export default UserAuctionListScreen;

const styles = StyleSheet.create({
  container: {
    flex: 1,
    paddingTop: 50,
    backgroundColor: '#ffffff'
  },
  header: {
    flexDirection: 'row'
```

User auction create screen

```
import React, { useEffect, useState } from 'react';      You, 2 days ago • auction create screen finalised layout
import {View, StyleSheet, Text, ScrollView, Image, TouchableOpacity} from 'react-native';
import { AppContainer } from '../container/container';
import themeColors from '../assets/colors';
import { AuctionBottomTab } from '../Navigations/AuctionBottomTab';
import { AppTextInput } from '../components/text_input.component';
import { AppButton } from '../components/button.component';
import { AppDatePicker } from '../components/datpicker.component';
import { MaterialIcons } from "@expo/vector-icons";

const UserAuctionCreateScreen = ({route, navigation}) => {
  const [image, setImage] = useState('https://via.placeholder.com/50');

  useEffect(() => {
    if (route.params === undefined) {
      setImage('https://via.placeholder.com/50')
    } else {
      setImage(route.params.image)
    }
  }, [])

  return (
    <View style={styles.container}>
      <AppContainer>
        <View style={styles.header}>
          <Text style={styles.auctionHeading}>Create Auction</Text>
        </View>
        <ScrollView>
          <View>
            <AppTextInput placeholder='Enter item name'/>
            <AppTextInput placeholder='Enter item price' keyboardType='numeric'/>
            <View style={styles.dateContainer}>
              <Text style={styles.label}>Expire Date</Text>
              <AppDatePicker/>
            </View>
            <Text style={styles.imgLabel}>Item Image</Text>
            <View style={styles.addImagewrapper}>
              <View style={styles.imageContainer}>
                <Image source={{
```

Home auction screen

```
import React from 'react';
import {View, StyleSheet, Text, ScrollView, Image} from 'react-native';
import { AppAuctionItem } from '../components/auction_item.component';
import { AppContainer } from '../container/container';
import themeColors from '../assets/colors';
import { AuctionBottomTab } from '../Navigations/AuctionBottomTab';

const HomeAuctionScreen = ({navigation}) => {
  return (
    <View style={styles.container}>
      <View style={styles.header}>
        <Image source={require('../assets/splash1.png')} style={styles.images}></Image>
      </View>
      <AppContainer>
        <Text style={styles.auctionHeading}>Auctions List</Text>
        <ScrollView>
          <View>
            <AppAuctionItem label='Auction Item 1' onPress = {() => navigation.navigate("AuctionDetailScreen")}/>
            <AppAuctionItem label='Auction Item 2'/>
            <AppAuctionItem label='Auction Item 3'/>
            <AppAuctionItem label='Auction Item 4'/>
          </View>
        </ScrollView>
      </AppContainer>
      <AuctionBottomTab/>
    </View>
  );
};
export default HomeAuctionScreen;

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#ffffff'
  },
  header: {
    height: '35%',
    marginBottom: 25,      You, 3 days ago • bottom tab added
    backgroundColor : themeColors.PRIMARY_COLOR
```

Auction List Screen

```
import React from 'react';      You, 3 days ago • auction screens created
import {View, StyleSheet, Text, ScrollView} from 'react-native';
import { AppAuctionItem } from '../components/auction_item.component';
import { AppContainer } from '../container/container';
import themeColors from '../assets/colors';
import { AuctionBottomTab } from '../Navigations/AuctionBottomTab';

const AuctionListScreen = ({navigation}) => {
  return (
    <View style={styles.container}>
      <AppContainer>
        <Text style={styles.auctionHeading}>Auctions List</Text>
        <ScrollView>
          <View>
            <AppAuctionItem label='Auction Item 1' onPress = {() => navigation.navigate("SignUp")}/>
            <AppAuctionItem label='Auction Item 2'/>
            <AppAuctionItem label='Auction Item 3'/>
            <AppAuctionItem label='Auction Item 4'/>
          </View>
        </ScrollView>
      </AppContainer>
      <AuctionBottomTab/>
    </View>
  );
};
export default AuctionListScreen;

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#ffffff'
  },
  header: {
    height: '35%',
    marginBottom: 25,
    backgroundColor : themeColors.PRIMARY_COLOR
  },
  btnRowContainer: {
    maxHeight: 100
```

Auction Detail Screen

```
import React from 'react';      You, 3 days ago • carousel component added to auction detail pae
import {View, StyleSheet, Text, ScrollView} from 'react-native';
import { AppStakeItem } from '../components/auction.stake.component';
import { AppContainer } from '../container/container';
import themeColors from '../assets/colors';
import { AuctionBottomTab } from '../Navigations/AuctionBottomTab';
import AppCarousel from '../components/carousel.component';
import { AppAddBid } from '../components/add_bid_button.component';

const AuctionDetailScreen = (props, {navigation}) => {
  return (
    <View style={styles.wrapper}>
      <View style={styles.header}>
        <AppCarousel/>
      </View>
      <View style={styles.containe
        <AppContainer>
          <View style={styles.titl
            <Text style={styles.au                    Item</Text>
            <View style={styles.lo                    ontainer}>
              <Text style={styles.auctionText}>Lot No : </Text>
              <View style={styles.lot}><Text style={styles.lotText}>17839</Text></View>
            </View>
          </View>
          <Text style={styles.auctionText}>Last Stakes</Text>
          <ScrollView horizontal style={{height: 100}}>
            <View style={styles.stakesContainer}>
              <AppStakeItem/>
              <AppStakeItem/>
              <AppStakeItem/>
              <AppStakeItem/>
            </View>
          </ScrollView>
          <AppAddBid/>
        </AppContainer>
        <AuctionBottomTab/>
      </View>
    </View>
  );

(property) auctionText: {
    marginTop: number;
    fontSize: number;
    fontWeight: "400";
    color: string;
}
```

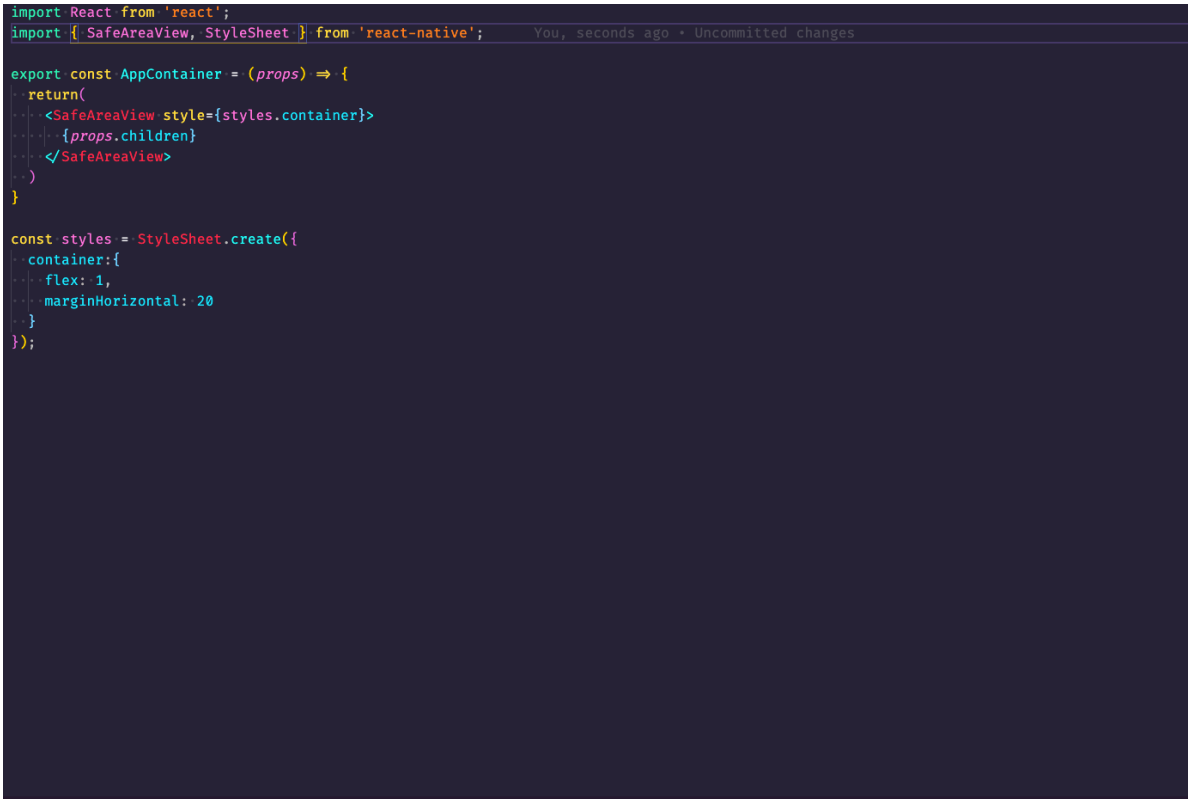
Auction Camera Screen

```
import React from 'react';      You, 2 days ago • auction create screen finalised layout
import { AppCamera } from '../components/camera.component';

const AuctionCameraScreen = ({navigation}) => {
  const handleAddImage = (image) => {
    navigation.navigate('AuctionCreateScreen', {
      image: image
    })
  }
  return (
    <AppCamera onAddImage={(image) => handleAddImage(image)}/>
  );
};
export default AuctionCameraScreen;
```

Auction Bottom Tab that is used for navigation

```
import React from "react";
import {
  StyleSheet,
  View,
  Text,       You, 3 days ago • bottom tab added
  TouchableOpacity,
} from "react-native";
import { MaterialIcons } from "@expo/vector-icons";
import { useNavigation } from '@react-navigation/native';
import themeColors from "../assets/colors";

export function AuctionBottomTab() {
    const navigation = useNavigation();
    return(
        <View style={styles.bottomBtn}>

            <TouchableOpacity style={styles.barButtonView}
                onPress={() => navigation.navigate("UserBidScreen")}
            >
            <MaterialIcons name="leaderboard" size={24} color={themeColors.WHITE} />
                <Text style={styles.iconText}>Bids</Text>
            </TouchableOpacity>

            <TouchableOpacity style={styles.barButtonView}
             onPress={() => navigation.navigate("UserAuctionScreen")}
            >
            <MaterialIcons name="gavel" size={24} color={themeColors.WHITE} />
                <Text style={styles.iconText}>Auctions</Text>
            </TouchableOpacity>

            <TouchableOpacity style={styles.barButtonView}
             onPress={() => navigation.navigate("AuctionHomeScreen")}
            >
            <MaterialIcons name="home" size={24} color={themeColors.WHITE} />
                <Text style={styles.iconText}>Home</Text>
            </TouchableOpacity>
        </View>
    );
}
```

App Container which is the wrapper for all screens to maintain consistency



```
import React from 'react';
import { SafeAreaView, StyleSheet } from 'react-native';      You, seconds ago • Uncommitted changes

export const AppContainer = (props) => {
  return(
    <SafeAreaView style={styles.container}>
      {props.children}
    </SafeAreaView>
  )
}

const styles = StyleSheet.create({
  container:{
    flex: 1,
    marginHorizontal: 20
  }
});
```

Custom Text Input component

```
import React from 'react';
import { StyleSheet, TextInput } from 'react-native';
import themeColors from '../assets/colors';

export const AppTextInput = (props) => {
  return(
    <TextInput style={styles.input} {...props}/>
  )
}

const styles = StyleSheet.create({
  input:{
    alignItems: "center",
    padding: 10,
    marginBottom: 20,      You, 2 days ago • auction create screen finalised layout
    fontSize: 16,
    borderBottomColor: themeColors.SECONDARY_COLOR,
    borderBottomWidth: 1,
  }
});
```

Custom app button component

```
import React from 'react';
import { StyleSheet, TouchableOpacity, Text } from 'react-native';
import themeColors from '../assets/colors';

export const AppButton = (props) => {      You, 3 days ago • button component created
  return(
    <TouchableOpacity onPress={props.onPress} style={styles.button}>
      <Text style={styles.label}>{props.label}</Text>
    </TouchableOpacity>
  )
}

const styles = StyleSheet.create({
  button:{
    alignItems: "center",
    padding: 10,
    fontSize: 16,
    backgroundColor: themeColors.PRIMARY_COLOR,
    borderRadius: 20,
    shadowColor: 'rgba(0,0,0, .4)',
    shadowOffset: { height: 2, width: 1 },
    shadowOpacity: 1,
    shadowRadius: 1,
    elevation: 2,
    marginVertical: 10
  },
  label: {
    fontSize: 20,
    color: themeColors.WHITE
  }
});
```

Custom date picker component

```
import React, {useState} from 'react';      You, 2 days ago • added datepicker component
import {View, StyleSheet, Platform} from 'react-native';
import DateTimePicker from '@react-native-community/datetimepicker';

export const AppDatePicker = () => {
  const [date, setDate] = useState(new Date());
  const [mode, setMode] = useState('date');
  const [show, setShow] = useState(false);

  const onChange = (event, selectedDate) => {
    const currentDate = selectedDate || date;
    setShow(Platform.OS === 'ios');
    setDate(currentDate);
  };

  const showMode = (currentMode) => {
    setShow(true);
    setMode(currentMode);
  };

  const showDatepicker = () => {
    showMode('date');
  };

  const showTimepicker = () => {
    showMode('time');
  };

  return (
    <View style={styles.datepicker}>
      <DateTimePicker
        testID="dateTimePicker"
        value={date}
        mode={mode}
        is24Hour={true}
        display="default"
        onChange={onChange}
      />
    </View>
```

Custom camera component

```
import React, { useState, useRef, useEffect } from 'react';      You, 2 days ago • auction create screen finalised layout
import {
  StyleSheet,
  Dimensions,
  View,
  Text,
  TouchableOpacity,
} from 'react-native';
import { Camera } from 'expo-camera';
import { AntDesign, MaterialIcons } from '@expo/vector-icons';
import themeColors from '../assets/colors';

const WINDOW_HEIGHT = Dimensions.get('window').height;
const CAPTURE_SIZE = Math.floor(WINDOW_HEIGHT * 0.08);

export const AppCamera = (props) => {
  const cameraRef = useRef();
  const [hasPermission, setHasPermission] = useState(null);
  const [cameraType, setCameraType] = useState(Camera.Constants.Type.back);
  const [isPreview, setIsPreview] = useState(false);
  const [isCameraReady, setIsCameraReady] = useState(false);

  useEffect(() => {
    onHandlePermission();
  }, []);

  const onHandlePermission = async () => {
    const { status } = await Camera.requestPermissionsAsync();
    setHasPermission(status === 'granted');
  };

  const onCameraReady = () => {
    setIsCameraReady(true);
  };

  const switchCamera = () => {
    if (isPreview) {
      return;
    }
```

Custom carousel component

```
import React, { Component } from 'react'
import { Animated, View, StyleSheet, Image, Dimensions, ScrollView , Text} from 'react-native'
import themeColors from '../assets/colors'

const deviceWidth = Dimensions.get('window').width
const FIXED_BAR_WIDTH = 100
const BAR_SPACE = 10

const images = [
  'https://s-media-cache-ak0.pinimg.com/originals/ee/51/39/ee5139157407967591081ee04723259a.png',
  'https://s-media-cache-ak0.pinimg.com/originals/40/4f/83/404f83e93175630e77bc29b3fe727cbe.jpg',
  'https://s-media-cache-ak0.pinimg.com/originals/8d/1a/da/8d1adab145a2d606c85e339873b9bb0e.jpg',
]

You, 2 days ago | 1 author (You)
export default class AppCarousel extends Component {

  numItems = images.length
  itemWidth = (FIXED_BAR_WIDTH / this.numItems) - ((this.numItems - 1) * BAR_SPACE)
  animVal = new Animated.Value(0)

  render() {
    let imageArray = []
    let barArray = []
    images.forEach((image, i) => {
      console.log(image, i)
      const thisImage = (
        <Image       You, 3 days ago • carousel component added to auction detail pae
          key={`image${i}`}
          source={{uri: image}}
          style={styles.image}
        />
      )
      imageArray.push(thisImage)

      const scrollBarVal = this.animVal.interpolate({
        inputRange: [deviceWidth * (i - 1), deviceWidth * (i + 1)],
        outputRange: [-this.itemWidth, this.itemWidth],
```

Custom Countdown timer component

```
import { CountdownCircleTimer } from 'react-native-countdown-circle-timer'
import React from 'react';
import {Animated} from 'react-native';
import themeColors from '../assets/colors';

const AppCountDownTimer = () => (
  <CountdownCircleTimer
    isPlaying={false}
    size={40}
    strokeWidth={5}
    duration={20}
    trailColor={themeColors.WHITE}      You, 2 days ago • count down component modified
    colors={[
      [themeColors.BLACK, 0.4],
      [themeColors.SECONDARY_COLOR, 0.4],
      [themeColors.RED, 0.2],
    ]}
  >
    {({ remainingTime }) => (
      <Animated.Text style={{ color: themeColors.WHITE }}>
        {remainingTime}
      </Animated.Text>
    )}
  </CountdownCircleTimer>
);

export default AppCountDownTimer
```

Custom list item for auction

```
import React from 'react';
import { StyleSheet, TouchableOpacity, Text, View } from 'react-native';
import themeColors from '../assets/colors';
import { MaterialIcons } from '@expo/vector-icons';
import AppCountDownTimer from './countdown_component';

export const AppAuctionItem = props => {
  return(
    <View>
    {props.delete ? (
      <View style={styles.container}>
        <TouchableOpacity onPress={props.onSelect} style={styles.button}>
          <Text style={styles.label}>{props.label}</Text>
          <AppCountDownTimer/>
        </TouchableOpacity>
        <TouchableOpacity onPress={props.onDelete} style={styles.delBtn}>
          <MaterialIcons name="delete" size={24} color={themeColors.WHITE} />
        </TouchableOpacity>
      </View>
    ) : (
      <View style={styles.container}>
        <TouchableOpacity onPress={props.onPress} style={styles.button}>
          <Text style={styles.label}>{props.label}</Text>
          <AppCountDownTimer/>
        </TouchableOpacity>
      </View>
    )}
    </View>

  )
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    flexDirection : 'row',
    justifyContent: 'space-between'
  },
  button:{
```

Custom icon button component

```
import React from 'react';
import { StyleSheet, TouchableOpacity, Text } from 'react-native';
import themeColors from '../assets/colors';

export const AppIconButton = (props) => {
  return(
    <TouchableOpacity onPress={props.onPress} style={styles.button}>
      <Text style={styles.label}>{props.label}</Text>
    </TouchableOpacity>
  )
}
      You, 3 days ago • auction welcome page updated
const styles = StyleSheet.create({
  button:{
    alignItems: "center",
    justifyContent: 'center',
    padding: 10,
    fontSize: 16,
    width: 150,
    height: 55,
    backgroundColor: themeColors.BLACK,
    borderRadius: 20,
    shadowColor: 'rgba(0,0,0, .4)',
    shadowOffset: { height: 2, width: 1 },
    shadowOpacity: 1,
    shadowRadius: 1,
    elevation: 2,
    marginHorizontal: 10
  },
  label: {
    fontSize: 20,
    color: themeColors.WHITE
  }
});
```

Custom stake item component to display bids

```
import React from 'react';      You, 3 days ago • stake item added to auction detail page
import { StyleSheet, View, Text } from 'react-native';
import themeColors from '../assets/colors';

export const AppStakeItem = (props) => {
  return(
    <View style={styles.stake}>
      <View style={styles.stakeWrapperEdges}/>
      <View style={styles.stakeSideEdges}>
        <View style={styles.circleTop}/>
          <View style={styles.dashedLine}/>
          <View style={styles.dashedLine}/>
          <View style={styles.dashedLine}/>
          <View style={styles.dashedLine}/>
        <View style={styles.circleEnd}/>
      </View>
      <View style={styles.stakeDetailsContent}>
        <View>
          <View  style={styles.lotTile}>
            <Text style={styles.lotText}>62</Text>
          </View>
        </View>
        <View style={styles.lotDetails}>
          <Text style={styles.lotName}>William Grey</Text>
          <Text style={styles.label}>Rs. 164.00</Text>
        </View>
      </View>
      <View style={styles.stakeSideEdges}>
        <View style={styles.circleTop}/>
          <View style={styles.dashedLine}/>
          <View style={styles.dashedLine}/>
          <View style={styles.dashedLine}/>
          <View style={styles.dashedLine}/>
        <View style={styles.circleEnd}/>
      </View>
      <View style={styles.stakeWrapperEdges}/>
    </View>
  )
}

(alias) class Text
import Text
```

Custom add bid component

```
import React from 'react';      You, 2 days ago • add auction component added
import { StyleSheet, TouchableOpacity, View, TextInput } from 'react-native';
import themeColors from '../assets/colors';
import { MaterialIcons} from "@expo/vector-icons";
export const AppAddBid = (props) => {
  return(
    <View style={styles.bidContainer}>
      <TextInput style={styles.input} keyboardType='number-pad' placeholder='Enter bid'/>
      <TouchableOpacity onPress={props.onPress} style={styles.bid}>
        <MaterialIcons name="add" size={16} color={themeColors.WHITE} />
      </TouchableOpacity>
    </View>
  )
}

const styles = StyleSheet.create({
  bidContainer: {
    flex: 1,
    marginVertical: 10,
    flexDirection: 'row',
    alignItems: 'center',
    justifyContent: 'space-between',
    paddingHorizontal: 20,
    borderRadius: 20,
    borderWidth: 1,
    borderColor: themeColors.TERTIARY_COLOR,
    maxHeight: 50
  },
  bid:{
    alignItems: "center",
    padding: 8,
    backgroundColor: themeColors.PRIMARY_COLOR,
    borderRadius: 20,
    shadowColor: 'rgba(0,0,0, .4)',
    shadowOffset: { height: 2, width: 1 },
    shadowOpacity: 1,
    shadowRadius: 1,
    elevation: 2,
    height: 33
```