app.py

```python
from flask import Flask, render_template
from flask_cors import CORS, cross_origin
import numpy as np
import pandas as pd
from datetime import datetime
import crops
import random

app = Flask(__name__)
app.config['CORS_HEADERS'] = 'Content-Type'

cors = CORS(app, resources={r"/ticker": {"origins": "http://localhost:port"}})

commodity_dict = {
    "arhar": "static/Arhar.csv",
    "bajra": "static/Bajra.csv",
    "barley": "static/Barley.csv",
    "copra": "static/Copra.csv",
    "cotton": "static/Cotton.csv",
    "sesamum": "static/Sesamum.csv",
    "gram": "static/Gram.csv",
    "groundnut": "static/Groundnut.csv",
    "jowar": "static/Jowar.csv",
    "maize": "static/Maize.csv",
    "masoor": "static/Masoor.csv",
    "moong": "static/Moong.csv",
    "niger": "static/Niger.csv",
    "paddy": "static/Paddy.csv",
    "ragi": "static/Ragi.csv",
    "rape": "static/Rape.csv",
    "jute": "static/Jute.csv",
    "safflower": "static/Safflower.csv",
    "soyabean": "static/Soyabean.csv",
    "sugarcane": "static/Sugarcane.csv",
    "sunflower": "static/Sunflower.csv",
    "urad": "static/Urad.csv",
    "wheat": "static/Wheat.csv"
}

annual_rainfall = [29, 21, 37.5, 30.7, 52.6, 150, 299, 251.7, 179.2, 70.5, 39.8,
10.9]
base = {
    "Paddy": 1245.5,
    "Arhar": 3200,
    "Bajra": 1175,
    "Barley": 980,
    "Copra": 5100,
    "Cotton": 3600,
    "Sesamum": 4200,
    "Gram": 2800,
    "Groundnut": 3700,
    "Jowar": 1520,
    "Maize": 1175,
    "Masoor": 2800,
    "Moong": 3500,
    "Niger": 3500,
    "Ragi": 1500,
    "Rape": 2500,
    "Jute": 1675,
    "Safflower": 2500,
    "Soyabean": 2200,
    "Sugarcane": 2250,
    "Sunflower": 3700,
    "Urad": 4300,
    "Wheat": 1350

}
commodity_list = []

class Commodity:

    def __init__(self, csv_name):
        self.name = csv_name
        dataset = pd.read_csv(csv_name)
        self.X = dataset.iloc[:, :-1].values
        self.Y = dataset.iloc[:, 3].values

        #from sklearn.model_selection import train_test_split
        #X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.1,
random_state=0)

        # Fitting decision tree regression to dataset
        from sklearn.tree import DecisionTreeRegressor
        self.regressor = DecisionTreeRegressor(max_depth=10,random_state=0)
        self.regressor.fit(self.X, self.Y)
        #y_pred_tree = self.regressor.predict(X_test)
        # fsa=np.array([float(1),2019,45]).reshape(1,3)
        # fask=regressor_tree.predict(fsa)

    def getPredictedValue(self, value):
        if value[1]>=2019:
            fsa = np.array(value).reshape(1, 3)
            #print(" ",self.regressor.predict(fsa)[0])
            return self.regressor.predict(fsa)[0]
        else:
            c=self.X[:,0:2]
            x=[]
            for i in c:
                x.append(i.tolist())
            fsa = [value[0], value[1]]
            ind = 0
            for i in range(0,len(x)):
                if x[i]==fsa:
                    ind=i
                    break
            #print(index, " ",ind)
            #print(x[ind])
            #print(self.Y[i])
            return self.Y[i]

    def getCropName(self):
        a = self.name.split('.')
        return a[0]

app = Flask(__name__)

@app.route('/')
def index():
    context = {
        "top5": TopFiveWinners(),
        "bottom5": TopFiveLosers(),
        "sixmonths": SixMonthsForecast()
    }
    return render_template('index.html')

@app.route('/commodity/<name>')
def crop_profile(name):
    max_crop, min_crop, forecast_crop_values = TwelveMonthsForecast(name)
    prev_crop_values = TwelveMonthPrevious(name)
    forecast_x = [i[0] for i in forecast_crop_values]
    forecast_y = [i[1] for i in forecast_crop_values]
    previous_x = [i[0] for i in prev_crop_values]
    previous_y = [i[1] for i in prev_crop_values]
    current_price = CurrentMonth(name)
    #print(max_crop)
    #print(min_crop)
    #print(forecast_crop_values)
    #print(prev_crop_values)
    #print(str(forecast_x))
    crop_data = crops.crop(name)
    context = {
        "name":name,
        "max_crop": max_crop,
        "min_crop": min_crop,
        "forecast_values": forecast_crop_values,
        "forecast_x": str(forecast_x),
        "forecast_y":forecast_y,
        "previous_values": prev_crop_values,
        "previous_x":previous_x,
        "previous_y":previous_y,
        "current_price": current_price,
        "image_url":crop_data[0],
        "prime_loc":crop_data[1],
        "type_c":crop_data[2],
        "export":crop_data[3]
    }
    return render_template('commodity.html', context=context)

def TopFiveWinners():
    current_month = datetime.now().month
    current_year = datetime.now().year
    current_rainfall = annual_rainfall[current_month - 1]
    prev_month = current_month - 1
    prev_rainfall = annual_rainfall[prev_month - 1]
    current_month_prediction = []
    prev_month_prediction = []
    change = []

    for i in commodity_list:
        current_predict = i.getPredictedValue([float(current_month), current_year,
current_rainfall])
        current_month_prediction.append(current_predict)
        prev_predict = i.getPredictedValue([float(prev_month), current_year,
prev_rainfall])
        prev_month_prediction.append(prev_predict)
        change.append((((current_predict - prev_predict) * 100 / prev_predict),
commodity_list.index(i)))
    sorted_change = change
    sorted_change.sort(reverse=True)
    # print(sorted_change)
    to_send = []
    for j in range(0, 5):
        perc, i = sorted_change[j]
        name = commodity_list[i].getCropName().split('/')[1]
        to_send.append([name, round((current_month_prediction[i] * base[name]) / 100,
2), round(perc, 2)])
    #print(to_send)
    return to_send


def TopFiveLosers():
    current_month = datetime.now().month
    current_year = datetime.now().year
    current_rainfall = annual_rainfall[current_month - 1]
    prev_month = current_month - 1
    prev_rainfall = annual_rainfall[prev_month - 1]
    current_month_prediction = []
    prev_month_prediction = []
    change = []

    for i in commodity_list:
        current_predict = i.getPredictedValue([float(current_month), current_year,
current_rainfall])
        current_month_prediction.append(current_predict)
        prev_predict = i.getPredictedValue([float(prev_month), current_year,
prev_rainfall])
        prev_month_prediction.append(prev_predict)
        change.append((((current_predict - prev_predict) * 100 / prev_predict),
commodity_list.index(i)))
    sorted_change = change
    sorted_change.sort()
    to_send = []
    for j in range(0, 5):
        perc, i = sorted_change[j]
        name = commodity_list[i].getCropName().split('/')[1]
        to_send.append([name, round((current_month_prediction[i] * base[name]) / 100,
2), round(perc, 2)])
   # print(to_send)
```

```
    return to_send



def SixMonthsForecast():
    month1=[]
    month2=[]
    month3=[]
    month4=[]
    month5=[]
    month6=[]
    for i in commodity_list:
        crop=SixMonthsForecastHelper(i.getCropName())
        k=0
        for j in crop:
            time = j[0]
            price = j[1]
            change = j[2]
            if k==0:
                month1.append((price,change,i.getCropName().split("/")[1],time))
            elif k==1:
                month2.append((price,change,i.getCropName().split("/")[1],time))
            elif k==2:
                month3.append((price,change,i.getCropName().split("/")[1],time))
            elif k==3:
                month4.append((price,change,i.getCropName().split("/")[1],time))
            elif k==4:
                month5.append((price,change,i.getCropName().split("/")[1],time))
            elif k==5:
                month6.append((price,change,i.getCropName().split("/")[1],time))
            k+=1
    month1.sort()
    month2.sort()
    month3.sort()
    month4.sort()
    month5.sort()
    month6.sort()
    crop_month_wise=[]
    crop_month_wise.append([month1[0][3],month1[len(month1)-1][2],month1[len(month1)-
1][0],month1[len(month1)-1][1],month1[0][2],month1[0][0],month1[0][1]])
    crop_month_wise.append([month2[0][3],month2[len(month2)-1][2],month2[len(month2)-
1][0],month2[len(month2)-1][1],month2[0][2],month2[0][0],month2[0][1]])
    crop_month_wise.append([month3[0][3],month3[len(month3)-1][2],month3[len(month3)-
1][0],month3[len(month3)-1][1],month3[0][2],month3[0][0],month3[0][1]])
    crop_month_wise.append([month4[0][3],month4[len(month4)-1][2],month4[len(month4)-
1][0],month4[len(month4)-1][1],month4[0][2],month4[0][0],month4[0][1]])
    crop_month_wise.append([month5[0][3],month5[len(month5)-1][2],month5[len(month5)-
1][0],month5[len(month5)-1][1],month5[0][2],month5[0][0],month5[0][1]])
    crop_month_wise.append([month6[0][3],month6[len(month6)-1][2],month6[len(month6)-
1][0],month6[len(month6)-1][1],month6[0][2],month6[0][0],month6[0][1]])

   # print(crop_month_wise)
    return crop_month_wise

def SixMonthsForecastHelper(name):
    current_month = datetime.now().month
    current_year = datetime.now().year
    current_rainfall = annual_rainfall[current_month - 1]
    name = name.split("/")[1]
    name = name.lower()
    commodity = commodity_list[0]
    for i in commodity_list:
        if name == str(i):
            commodity = i
            break
    month_with_year = []
    for i in range(1, 7):
        if current_month + i <= 12:
            month_with_year.append((current_month + i, current_year,
annual_rainfall[current_month + i - 1]))
        else:
            month_with_year.append((current_month + i - 12, current_year + 1,
annual_rainfall[current_month + i - 13]))
    wpis = []
    current_wpi = commodity.getPredictedValue([float(current_month), current_year,
current_rainfall])
    change = []

    for m, y, r in month_with_year:
        current_predict = commodity.getPredictedValue([float(m), y, r])
        wpis.append(current_predict)
        change.append(((current_predict - current_wpi) * 100) / current_wpi)

    crop_price = []
    for i in range(0, len(wpis)):
        m, y, r = month_with_year[i]
        x = datetime(y, m, 1)
        x = x.strftime("%b %y")
        crop_price.append([x, round((wpis[i]* base[name.capitalize()]) / 100, 2) ,
round(change[i], 2)])

   # print("Crop_Price: ", crop_price)
    return crop_price

def CurrentMonth(name):
    current_month = datetime.now().month
    current_year = datetime.now().year
    current_rainfall = annual_rainfall[current_month - 1]
    name = name.lower()
    commodity = commodity_list[0]
    for i in commodity_list:
        if name == str(i):
            commodity = i
            break
    current_wpi = commodity.getPredictedValue([float(current_month), current_year,
current_rainfall])
    current_price = (base[name.capitalize()]*current_wpi)/100
    return current_price

def TwelveMonthsForecast(name):
    current_month = datetime.now().month
    current_year = datetime.now().year
    current_rainfall = annual_rainfall[current_month - 1]
    name = name.lower()
    commodity = commodity_list[0]
    for i in commodity_list:
        if name == str(i):
            commodity = i
            break
    month_with_year = []
    for i in range(1, 13):
        if current_month + i <= 12:
            month_with_year.append((current_month + i, current_year,
annual_rainfall[current_month + i - 1]))
        else:
            month_with_year.append((current_month + i - 12, current_year + 1,
annual_rainfall[current_month + i - 13]))
    max_index = 0
    min_index = 0
    max_value = 0
    min_value = 9999
    wpis = []
    current_wpi = commodity.getPredictedValue([float(current_month), current_year,
current_rainfall])
    change = []

    for m, y, r in month_with_year:
        current_predict = commodity.getPredictedValue([float(m), y, r])
        if current_predict > max_value:
            max_value = current_predict
            max_index = month_with_year.index((m, y, r))
        if current_predict < min_value:
            min_value = current_predict
            min_index = month_with_year.index((m, y, r))
        wpis.append(current_predict)
        change.append(((current_predict - current_wpi) * 100) / current_wpi)

    max_month, max_year, r1 = month_with_year[max_index]
    min_month, min_year, r2 = month_with_year[min_index]
    min_value = min_value * base[name.capitalize()] / 100
    max_value = max_value * base[name.capitalize()] / 100
    crop_price = []
    for i in range(0, len(wpis)):
        m, y, r = month_with_year[i]
        x = datetime(y, m, 1)
        x = x.strftime("%b %y")
        crop_price.append([x, round((wpis[i]* base[name.capitalize()]) / 100, 2) ,
round(change[i], 2)])
   # print("forecasr", wpis)
    x = datetime(max_year,max_month,1)
    x = x.strftime("%b %y")
    max_crop = [x, round(max_value,2)]
    x = datetime(min_year, min_month, 1)
    x = x.strftime("%b %y")
    min_crop = [x, round(min_value,2)]
    return max_crop, min_crop, crop_price


def TwelveMonthPrevious(name):
    name = name.lower()
    current_month = datetime.now().month
    current_year = datetime.now().year
    current_rainfall = annual_rainfall[current_month - 1]
    commodity = commodity_list[0]
    wpis = []
    crop_price = []
    for i in commodity_list:
        if name == str(i):
            commodity = i
            break
    month_with_year = []
    for i in range(1, 13):
        if current_month - i >= 1:
            month_with_year.append((current_month - i, current_year,
annual_rainfall[current_month - i - 1]))
        else:
            month_with_year.append((current_month - i + 12, current_year - 1,
annual_rainfall[current_month - i + 11]))

    for m, y, r in month_with_year:
        current_predict = commodity.getPredictedValue([float(m), 2013, r])
        wpis.append(current_predict)

    for i in range(0, len(wpis)):
        m, y, r = month_with_year[i]
        x = datetime(y,m,1)
        x = x.strftime("%b %y")
        crop_price.append([x, round((wpis[i]* base[name.capitalize()]) / 100, 2)])
   # print("previous ", wpis)
    new_crop_price =[]
    for i in range(len(crop_price)-1,-1,-1):
        new_crop_price.append(crop_price[i])
    return new_crop_price



if __name__ == "__main__":
    arhar = Commodity(commodity_dict["arhar"])
    commodity_list.append(arhar)
    bajra = Commodity(commodity_dict["bajra"])
    commodity_list.append(bajra)
    barley = Commodity(commodity_dict["barley"])
    commodity_list.append(barley)
    copra = Commodity(commodity_dict["copra"])
    commodity_list.append(copra)
    cotton = Commodity(commodity_dict["cotton"])
    commodity_list.append(cotton)
    sesamum = Commodity(commodity_dict["sesamum"])
    commodity_list.append(sesamum)
    gram = Commodity(commodity_dict["gram"])
    commodity_list.append(gram)
    groundnut = Commodity(commodity_dict["groundnut"])
    commodity_list.append(groundnut)
    jowar = Commodity(commodity_dict["jowar"])
    commodity_list.append(jowar)
    maize = Commodity(commodity_dict["maize"])
    commodity_list.append(maize)
    masoor = Commodity(commodity_dict["masoor"])
    commodity_list.append(masoor)
    moong = Commodity(commodity_dict["moong"])
    commodity_list.append(moong)
    niger = Commodity(commodity_dict["niger"])
    commodity_list.append(niger)
    paddy = Commodity(commodity_dict["paddy"])
    commodity_list.append(paddy)
    ragi = Commodity(commodity_dict["ragi"])
    commodity_list.append(ragi)
    rape = Commodity(commodity_dict["rape"])
    commodity_list.append(rape)
    jute = Commodity(commodity_dict["jute"])
    commodity_list.append(jute)
    safflower = Commodity(commodity_dict["safflower"])
    commodity_list.append(safflower)
    soyabean = Commodity(commodity_dict["soyabean"])
    commodity_list.append(soyabean)
    sugarcane = Commodity(commodity_dict["sugarcane"])
    commodity_list.append(sugarcane)
    sunflower = Commodity(commodity_dict["sunflower"])
    commodity_list.append(sunflower)
    urad = Commodity(commodity_dict["urad"])
    commodity_list.append(urad)
    wheat = Commodity(commodity_dict["wheat"])
    commodity_list.append(wheat)

    app.run(debug=True)
```

accuracy.py

```
from sklearn.model_selection import StratifiedKFold,KFold
import pandas as pd
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor


def get_score_DTR(X_train, X_test, Y_train, Y_test):
    regressor = DecisionTreeRegressor(max_depth=4)
    regressor.fit(X_train, Y_train)
    return regressor.score(X_train, Y_train)

def get_score_RFR(X_train, X_test, Y_train, Y_test):
    regressor = RandomForestRegressor(max_depth=4)
    regressor.fit(X_train, Y_train)
    return regressor.score(X_train, Y_train)
```

```
commodity_list = [
    "static/Arhar.csv",
    "static/Bajra.csv",
    "static/Barley.csv",
    "static/Copra.csv",
    "static/Cotton.csv",
    "static/Sesamum.csv",
    "static/Gram.csv",
    "static/Groundnut.csv",
    "static/Jowar.csv",
    "static/Maize.csv",
    "static/Masoor.csv",
    "static/Moong.csv",
    "static/Niger.csv",
    "static/Paddy.csv",
    "static/Ragi.csv",
    "static/Rape.csv",
    "static/Jute.csv",
    "static/Safflower.csv",
    "static/Soyabean.csv",
    "static/Sugarcane.csv",
    "static/Sunflower.csv",
    "static/Urad.csv",
    "static/Wheat.csv"
]

acc_dtr=[]
acc_rfr=[]

for datafile in commodity_list:
    dataset = pd.read_csv(datafile)
    X = dataset.iloc[:, :-1].values
    Y = dataset.iloc[:, 3].values

    kf = KFold(n_splits=10)
    sc_dtr = []
    sc_rfr = []
    for train_index,test_index in kf.split(dataset):
        X_train, X_test, Y_train, Y_test = X[train_index], X[test_index],
Y[train_index], Y[test_index]

        sc_dtr.append(get_score_DTR(X_train, X_test, Y_train, Y_test))
        sc_rfr.append(get_score_RFR(X_train, X_test, Y_train, Y_test))
    acc_dtr.append(sum(sc_dtr)/len(sc_dtr))
    acc_rfr.append(sum(sc_rfr)/len(sc_rfr))


avg_dtr=sum(acc_dtr)/len(acc_dtr)
avg_rfr=sum(acc_rfr)/len(acc_rfr)

print ("\nThe accuracy for Decision Tree Regressor is",avg_dtr,"%.")
print ("\nThe accuracy for Random Forest Regressor is",avg_rfr,"%.")
```

crops.py

```
def crop(crop_name):
    crop_data = {
    "wheat":["/static/images/wheat.jpg", "U.P., Punjab, Haryana, Rajasthan, M.P.,
bihar", "rabi","Sri Lanka, United Arab Emirates, Taiwan"],
    "paddy":["/static/images/paddy.jpg", "W.B., U.P., Andhra Pradesh, Punjab, T.N.",
"kharif","Bangladesh, Saudi Arabia, Iran"],
    "barley":["/static/images/barley.jpg", "Rajasthan, Uttar Pradesh, Madhya Pradesh,
Haryana, Punjab", "rabi","Oman, UK, Qatar, USA"],
    "maize":["/static/images/maize.jpg", "Karnataka, Andhra Pradesh, Tamil Nadu,
Rajasthan, Maharashtra", "kharif", "Hong Kong, United Arab Emirates, France"],
    "bajra":["/static/images/bajra.jpg", "Rajasthan, Maharashtra, Haryana, Uttar
Pradesh and Gujarat", "kharif", "Oman, Saudi Arabia, Israel, Japan"],
    "copra":["/static/images/copra.jpg", "Kerala, Tamil Nadu, Karnataka, Andhra
Pradesh, Orissa, West Bengal","rabi", "Veitnam, Bangladesh, Iran, Malaysia"],
    "cotton":["/static/images/cotton.jpg", "Punjab, Haryana, Maharashtra, Tamil Nadu,
Madhya Pradesh, Gujarat", " China, Bangladesh, Egypt"],
    "masoor":["/static/images/masoor.jpg", "Uttar Pradesh, Madhya Pradesh, Bihar,
West Bengal, Rajasthan", "rabi", "Pakistan, Cyprus,United Arab Emirates"],
    "gram":["/static/images/gram.jpg", "Madhya Pradesh, Maharashtra, Rajasthan, Uttar
Pradesh, Andhra Pradesh & Karnataka", "rabi", "Veitnam, Spain, Myanmar"],
    "groundnut":["/static/images/groundnut.jpg", "Andhra Pradesh, Gujarat, Tamil
Nadu, Karnataka, and Maharashtra", "kharif", "Indonesia, Jordan, Iraq"],
    "arhar":["/static/images/arhar.jpg", "Maharashtra, Karnataka, Madhya Pradesh and
Andhra Pradesh", "kharif", "United Arab Emirates, USA, Chicago"],
    "sesamum":["/static/images/sesamum.jpg", "Maharashtra, Rajasthan, West Bengal,
Andhra Pradesh, Gujarat", "rabi", "Iraq, South Africa, USA, Netherlands"],
    "jowar":["/static/images/jowar.jpg", "Maharashtra, Karnataka, Andhra Pradesh,
Madhya Pradesh, Gujarat", "kharif", "Torronto, Sydney, New York"],
    "moong":["/static/images/moong.jpg", "Rajasthan, Maharashtra, Andhra Pradesh",
"rabi", "Qatar, United States, Canada"],
    "niger":["/static/images/niger.jpg", "Andha Pradesh, Assam, Chattisgarh, Gujarat,
Jharkhand", "kharif", "United States of American,Argenyina, Belgium"],
    "rape":["/static/images/rape.jpg", "Rajasthan, Uttar Pradesh, Haryana, Madhya
Pradesh, and Gujarat", "rabi", "Veitnam, Malaysia, Taiwan"],
    "jute":["/static/images/jute.jpg", " West Bengal , Assam , Orissa , Bihar , Uttar
Pradesh", "kharif", "JOrdan, United Arab Emirates, Taiwan"],
    "safflower":["/static/images/safflower.jpg",  "Maharashtra, Karnataka, Andhra
Pradesh, Madhya Pradesh, Orissa", "kharif", " Philippines, Taiwan, Portugal"],
    "soyabean":["/static/images/soyabean.jpg",  "Madhya Pradesh, Maharashtra,
Rajasthan, Madhya Pradesh and Maharashtra", "kharif", "Spain, Thailand, Singapore"],
    "urad":["/static/images/urad.jpg",  "Andhra Pradesh, Maharashtra, Madhya Pradesh,
Tamil Nadu", "rabi", "United States, Canada, United Arab Emirates"],
    "ragi":["/static/images/ragi.jpg",  "Maharashtra, Tamil Nadu and Uttarakhand",
"kharif", "United Arab Emirates, New Zealand, Bahrain"],
    "sunflower":["sunflower.jpg",  "Karnataka, Andhra Pradesh, Maharashtra, Bihar,
Orissa", "rabi", "Phillippines, United States, Bangladesh"],
    "sugarcane":["sugarcane.jpg","Uttar Pradesh, Maharashtra, Tamil Nadu, Karnataka,
Andhra Pradesh" , "kharif", "Kenya, United Arab Emirates, United Kingdom"]
    }
    return crop_data[crop_name]
```

commodity.html

```
<!DOCTYPE html>
<html>
<head>
```

```
  <title>Crop Price Prediction</title>
  <!-- Compiled and minified CSS -->
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.mi
n.css">
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">

    <!-- Compiled and minified JavaScript -->
     <style>
        div.main {
            padding: 5px 50px 75px 50px;
        }
    </style>

</head>
<body >
  <div class="main">
    <div class="nav-wrapper">

    <h3 class="card-panel #ffd54f amber lighten-2 center valign-wrapper
center"><a class="brand-logo  #ffd54f-text text-lighten-4" href="#"><img alt=""
src="/static/ApnaAnaajLogo.png" height="100px" width="100px" class="responsive-
img" />
          <span>Crop Price Prediction</span></a>
          </a></h3><h3>

  </div>
    <h2 class="header">{{context.name}}</h2>
    <div class="row">
        <div class="col s8 m7">
      <div class="card horizontal medium">
        <div class="card-image">
          <img src={{context.image_url}}>
        </div>
        <div class="card-stacked">
          <div class="card-content">
            <table>
              <tr>
                <td>Current Price</td>
                <td><b>$ {{context.current_price}} / Ton</b></td>
              </tr>
              <tr>
                <td>Crop Type</td>
                <td><b>{{context.type_c}}</b></td>
              </tr>
              <tr>
                <td>Export</td>
                <td><b>{{context.export}}</b></td>
              </tr>
            </table>
          </div>
        </div>
      </div>
    </div>

    <div class="col s4">
      <div class="card grey lighten-3">
        <div class="card-content black-text">
           <span class="card-title">Brief Forecast</span>
          <table>
            <tr>
              <td><p>Min. crop price time</p>
              <td><h5>{{context.min_crop[0]}}</h5></td>
              <td>
                <h4>${{context.min_crop[1]}}</h4>
              </td>
            </tr>
            <tr>
              <td><p>Max. crop price time</p>
              <td><h5>{{context.max_crop[0]}}</h5></td>
              <td>
                <h4>${{context.max_crop[1]}}</h4>
              </td>
            </tr>
          </table>

        </div>
      </div>
    </div>
  </div>

</div>
  <div class="row">
    <div class="col s4">
      <h5>Forecast Trends</h5>
     <table class="striped">
        <thead>
          <tr>
              <th>Month</th>
              <th>Price (per Qtl.)</th>
              <th>Change</th>
          </tr>
        </thead>

        <tbody>
        {% for item in context.forecast_values %}
          <tr>
            <td>{{item[0]}}</td>
            <td>${{item[1]}}</td>
            <td class="valign-wrapper">{{item[2]}}% {% if item[2]>=0 %}<img src=
"../static/gain-icon.png" height="25" width="25">{% else %}<img src=
"../static/loss-icon.png" height="25" width="25">{% endif %}</td>
          </tr>
        {% endfor %}
        </tbody>
      </table>
</div>
<div class="col s2"></div>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.4.0/Chart.min.js"></script
>

    <div class="chartjs-wrapper col s7 offset-s1">
    <canvas id="chartjs-0" class="chartjs" width="undefined" height="undefined"
style=" width:800px !important;
  height:400px !important;">
    </canvas>
  <script>
    new Chart(
    document.getElementById("chartjs-
0"),{"type":"line","data":{"labels":{{context.forecast_x|safe}},"datasets":[{"lab
el":"Next year
Price","data":{{context.forecast_y}},"fill":false,"borderColor":"rgb(75, 192,
192)","lineTension":0.1}]},"options":{ responsive: true,
    maintainAspectRatio: false,
    scales: {
        yAxes: [{
            display: true,
            ticks: {
                suggestedMin: 1000,
                stepSize : 200
            }
        }]
    }}});
    </script>
  </div>
  <div class="chartjs-wrapper col s7 offset-s1">
    <canvas id="chartjs-1" class="chartjs" width="undefined" height="undefined"
style=" width:800px !important;
  height:400px !important;">
    </canvas>
  <script>
    new Chart(document.getElementById("chartjs-
1"),{"type":"line","data":{"labels":{{context.previous_x|safe}},"datasets":[{"lab
el":"Previous year
price","data":{{context.previous_y}},"fill":false,"borderColor":"rgb(75, 192,
192)","lineTension":0.1}]},"options":{ responsive: true,
    maintainAspectRatio: false,
    scales: {
        yAxes: [{
            display : true,
            ticks: {
                suggestedMin : 1000,
                stepSize : 200
            }
        }]
    }}});</script>
  </div>
  </div>
</script>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.
js"></script>
</body>
</html>
```

Index.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Crop Price Prediction</title>
  <!-- Compiled and minified CSS -->
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.mi
n.css">
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">


    <!-- Compiled and minified JavaScript -->
    <style>
        div.main {
            padding: 5px 50px 75px 50px;
        }

        span,img {
            vertical-align: middle;
         }

        .brand-logo > span,.brand-logo > img {
            vertical-align: middle;
        }


    </style>
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

</head>

    <body class="main">
        <div>
            <div class="nav-wrapper">
```

```
                <h3 class="card-panel #ffd54f amber lighten-2 center valign-
wrapper center"><a class="brand-logo  #ffd54f-text text-lighten-4" href="#"><img
alt="" src="/static/ApnaAnaajLogo.png" height="100px" width="100px"
class="responsive-img" />
                <span>Crop Price Prediction</span></a>
                </a></h3><h3>

            </div>

            <h5>Explore by commodity</h5>
    <div class="row">
        <div class="col s3">
        <a href="http://localhost:5000/commodity/paddy" style="color: #000000">
        <div class="card grey lighten-4">
            <div class="card-content row valign-wrapper">
            <div class="col s3">
            <img src="https://img.icons8.com/color/48/000000/rice-bowl.png">
            </div>
            <div class="col s9">
            <span class="card-title">Rice</span>
            </div>
            </div>
        </div>
        </a>
        </div>
        <div class="col s3">
        <a href="http://localhost:5000/commodity/wheat" style="color: #000000">
        <div class="card grey lighten-4">
            <div class="card-content row valign-wrapper">
            <div class="col s3">
            <img src="https://img.icons8.com/color/48/000000/wheat.png">
            </div>
            <div class="col s9">
            <span class="card-title">Wheat</span>
            </div>
            </div>
        </div>
        </a>
        </div> <div class="col s3">
        <a href="http://localhost:5000/commodity/barley" style="color: #000000">
        <div class="card grey lighten-4">
            <div class="card-content row valign-wrapper">
            <div class="col s3">
            <img src="https://img.icons8.com/color/48/000000/barley.png">
            </div>
            <div class="col s9">
            <span class="card-title">Barley</span>
            </div>
            </div>
        </div>
        </a>
        </div> <div class="col s3">
        <a href="http://localhost:5000/commodity/soyabean" style="color:
#000000">
        <div class="card grey lighten-4">
            <div class="card-content row valign-wrapper">
            <div class="col s3">
            <img src="https://img.icons8.com/color/48/000000/soy.png">
            </div>
            <div class="col s9">
            <span class="card-title">Soybean</span>
            </div>
            </div>
        </div>
        </a>
        </div>
    </div>
        <div class="row">
        <div class="col s3">
        <a href="http://localhost:5000/commodity/cotton" style="color: #000000">
        <div class="card grey lighten-4">
            <div class="card-content row valign-wrapper">
            <div class="col s3">
            <img src="https://img.icons8.com/color/48/000000/cotton.png">
            </div>
            <div class="col s9">
            <span class="card-title">Cotton</span>
            </div>
            </div>
        </div>
        </a>
        </div>
        <div class="col s3">
        <a href="http://localhost:5000/commodity/copra" style="color: #000000">
        <div class="card grey lighten-4">
            <div class="card-content row valign-wrapper">
            <div class="col s3">
            <img src="https://img.icons8.com/color/48/000000/coconut.png">
            </div>
            <div class="col s9">
            <span class="card-title">Coconut</span>
            </div>
            </div>
        </div>
        </a>
        </div> <div class="col s3">
        <a href="http://localhost:5000/commodity/groundnut" style="color:
#000000">
        <div class="card grey lighten-4">
            <div class="card-content row valign-wrapper">
            <div class="col s3">
            <img src="https://img.icons8.com/color/48/000000/peanuts.png">
            </div>
            <div class="col s9">
            <span class="card-title">peanuts</span>
            </div>
            </div>
        </div>
        </a>
        </div> <div class="col s3">
        <a href="http://localhost:5000/commodity/rape" style="color: #000000">
        <div class="card grey lighten-4">
            <div class="card-content row valign-wrapper">
            <div class="col s3">
            <img src="https://image.flaticon.com/icons/svg/188/188317.svg">
            </div>
            <div class="col s9">
            <span class="card-title">Mustard</span>
            </div>
            </div>
        </div>
        </a>
        </div>
    </div>
        <div class="row">
        <div class="col s3">
        <a href="http://localhost:5000/commodity/sesamum" style="color: #000000">
        <div class="card grey lighten-4">
            <div class="card-content row valign-wrapper">
            <div class="col s3">
            <img src="https://img.icons8.com/color/48/000000/sesame.png">
            </div>
            <div class="col s9">
            <span class="card-title">sesame</span>
            </div>
            </div>
        </div>
        </a>
        </div>
        <div class="col s3">
        <a href="http://localhost:5000/commodity/gram" style="color: #000000">
        <div class="card grey lighten-4">
            <div class="card-content row valign-wrapper">
            <div class="col s3">
            <img src="https://img.icons8.com/color/48/000000/paper-bag-with-
seeds.png">
            </div>
            <div class="col s9">
            <span class="card-title">Chickpea</span>
            </div>
            </div>
        </div>
        </a>
        </div> <div class="col s3">
        <a href="http://localhost:5000/commodity/sugarcane" style="color:
#000000">
```

```
        <div class="card grey lighten-4">
            <div class="card-content row valign-wrapper">
            <div class="col s3">
            <img src="https://img.icons8.com/color/48/000000/sugarcane.png">
            </div>
            <div class="col s9">
            <span class="card-title">Sugarcane</span>
            </div>
            </div>
        </div>
        </a>
        </div> <div class="col s3">
        <a href="http://localhost:5000/commodity/arhar" style="color: #000000">
        <div class="card grey lighten-4">
            <div class="card-content row valign-wrapper">
            <div class="col s3">
            <img src="https://img.icons8.com/color/48/000000/paper-bag-with-
seeds.png">
            </div>
            <div class="col s9">
            <span class="card-title">Pigeon Pea</span>
            </div>
            </div>
        </div>
        </a>
        </div>
    </div>
        <div class="row">
        <div class="col s3">
        <a href="http://localhost:5000/commodity/ragi" style="color: #000000">
        <div class="card grey lighten-4">
            <div class="card-content row valign-wrapper">
            <div class="col s3">
            <img src="https://img.icons8.com/office/48/000000/wheat.png">
            </div>
            <div class="col s9">
            <span class="card-title">Finger Millet</span>
            </div>
            </div>
        </div>
        </a>
        </div>
        <div class="col s3">
        <a href="http://localhost:5000/commodity/maize" style="color: #000000">
        <div class="card grey lighten-4">
            <div class="card-content row valign-wrapper">
            <div class="col s3">
            <img src="https://img.icons8.com/color/48/000000/corn.png">
            </div>
            <div class="col s9">
            <span class="card-title">Maize</span>
            </div>
```

```html
            </div>
        </div>
        </a>
        </div> <div class="col s3">
        <a href="http://localhost:5000/commodity/moong" style="color: #000000">
        <div class="card grey lighten-4">
            <div class="card-content row valign-wrapper">
            <div class="col s3">
            <img src="https://img.icons8.com/color/48/000000/paper-bag-with-
seeds.png">
            </div>
            <div class="col s9">
            <span class="card-title">Green Gram</span>
            </div>
            </div>
        </div>
        </a>
        </div> <div class="col s3">
        <a href="http://localhost:5000/commodity/masoor" style="color: #000000">
        <div class="card grey lighten-4">
            <div class="card-content row valign-wrapper">
            <div class="col s3">
            <img src="https://img.icons8.com/color/48/000000/paper-bag-with-
seeds.png">
            </div>
            <div class="col s9">
            <span class="card-title">Red Lentil</span>
            </div>
            </div>
        </div>
        </a>
        </div>
    </div>
        <div class="row">
        <div class="col s3">
        <a href="http://localhost:5000/commodity/urad" style="color: #000000">
        <div class="card grey lighten-4">
            <div class="card-content row valign-wrapper">
            <div class="col s3">
            <img src="https://img.icons8.com/color/48/000000/paper-bag-with-
seeds.png">
            </div>
            <div class="col s9">
            <span class="card-title">Black Gram</span>
            </div>
            </div>
        </div>
        </a>
        </div>
        <div class="col s3">
        <a href="http://localhost:5000/commodity/jute" style="color: #000000">
        <div class="card grey lighten-4">
```

```
            <div class="card-content row valign-wrapper">
            <div class="col s3">
            <img src="https://img.icons8.com/color/48/000000/potato.png">
            </div>
            <div class="col s9">
            <span class="card-title">Raw Jute</span>
            </div>
            </div>
        </div>
        </a>
        </div> <div class="col s3">
        <a href="http://localhost:5000/commodity/niger" style="color: #000000">
        <div class="card grey lighten-4">
            <div class="card-content row valign-wrapper">
            <div class="col s3">
            <img src="https://img.icons8.com/color/48/000000/paper-bag-with-
seeds.png">
            </div>
            <div class="col s9">
            <span class="card-title">Niger Seed</span>
            </div>
            </div>
        </div>
        </a>
        </div> <div class="col s3">
        <a href="http://localhost:5000/commodity/safflower" style="color:
#000000">
        <div class="card grey lighten-4">
            <div class="card-content row valign-wrapper">
            <div class="col s3">
            <img src="https://img.icons8.com/color/48/000000/paper-bag-with-
seeds.png">
            </div>
            <div class="col s9">
            <span class="card-title">Safflower Seed</span>
            </div>
            </div>
        </div>
        </a>
        </div>
    </div>
        <div class="row">
        <div class="col s3">
        <a href="http://localhost:5000/commodity/sunflower" style="color:
#000000">
        <div class="card grey lighten-4">
            <div class="card-content row valign-wrapper">
            <div class="col s3">
            <img src="https://img.icons8.com/color/48/000000/sun.png">
            </div>
            <div class="col s9">
            <span class="card-title">Sunflower</span>
```

```html
                </div>
                </div>
            </div>
            </a>
            </div>
            <div class="col s3">
            <a href="http://localhost:5000/commodity/jowar" style="color: #000000">
            <div class="card grey lighten-4">
                <div class="card-content row valign-wrapper">
                <div class="col s3">
                <img src="https://img.icons8.com/color/48/000000/potato.png">
                </div>
                <div class="col s9">
                <span class="card-title">Sorghum</span>
                </div>
                </div>
            </div>
            </a>
            </div> <div class="col s3">
            <a href="http://localhost:5000/commodity/bajra" style="color: #000000">
            <div class="card grey lighten-4">
                <div class="card-content row valign-wrapper">
                <div class="col s3">
                <img src="https://img.icons8.com/color/48/000000/potato.png">
                </div>
                <div class="col s9">
                <span class="card-title">Pearl Millet</span>
                </div>
                </div>
            </div>
            </a>
            </div>
        </div>

        </body>

    </html>
```