CRUDE OIL PRICE PREDICTION

Submitted by

Team ID	**PNT2022TMID23859**
Project Name	Crude Oil Price Prediction
612919103110	YALINI C
612919103083	RISHITA A
612919103112	YESHWANTHINI J
612919103085	ROUPA DHARSHINI R K

VIVEKANANDHA
COLLEGE OF ENGINEERING FOR WOMEN

[Autonomous]

Approved by AICTE, New Delhi andAccreditedby NBA Affiliated to Anna University, Chennai-25,

An ISO 9001-2008Certified Institution

Elayampalayam, Tiruchengode,
Namakkal Dt. – 637205

NOVEMBER & 2022

ELECTRONICS AND COMMUNICATION ENGINEERING

TABLE OF CONTENTS

1 INTRODUCTION

Crude oil is the most important resources in today's world.Cost has the direct effect on the global habitat , our economy ,political and other activities.Here the rise in price of oil will profit for the producers. The evaporation nature of crude oil , and its price prediction become extremely difficult tasks. we are here proposing the innovative method of predicting the crude oil price using the artificial neural network(ANN). These proposed system helps us to decide the proper crude oil price at correct time analysis.

1.1 OBJECTIVES:

BY THE END OF PROJECT WE WILL BE ABLE TO:

- Using the flask frame work we will be able to build an application.
- Understanding of time series data.
- Techniques and concepts of time series forecasting.
- Splits the data and analysis.

1.2 PROJECT FLOW:

TO COMPLETE CERTAIN ACTIVITIES AND TO ACCOMPLISH THE TASKS LISTED BELOW:

- Collection of data
- Dataset creation

- Preprocessing of data
- Libraries imports
- Dataset imports
- Data analyze
- Missing data
- Featuring the data scaling
- Visualizaton of the dataset and analyzing
- Splitting the data into test and train
- Sliding window of data set
- Model building
- Model building libraries importation
- Initialize the model
- Adding LSTM and outline layers
- Learning model and training the process
- Evaluation of model
- Save the model
- Test the model
- Building application
- Create the HTML file
- Build the python code

2 LITERATURE SURVEY

2.1 Existing System

Crude oil is amongst the most important resources in today's world, it is the chief fuel and its cost has a direct effect on the global habitat, our economy and oil exploration, exploitation and other activities. Prediction of oil prices has become the need of the hour, it is a boon to many large and small industries, individuals, the government. The evaporative nature of crude oil, its price prediction becomes extremely difficult and it is hard to be precise with the same. Several different factors that affect crude oil prices. We propose a contemporary and innovative method of predicting crude oil prices using the artificial neural network (ANN). The main advantage of this approach of ANN is that it continuously captures the unstable pattern of the crude oil prices which have been incorporated by finding out the optimal lag and number of the delay effect that controls the prices of crude oil. Variation of lag in a period of time has been done for the most optimum and close results, we then have validated our results by evaluating the root mean square error and the results obtained using the proposed model have significantly outperformed.

2.2 REFERENCES

HarunaChiroma et.al in Evolutionary Neural Network model for West Texas Intermediate crude oil price prediction :

Has observed and provided,
an alternative approach based on a genetic algorithm and neural network (GA–NN) for the prediction of the West Texas Intermediate (WTI) crude oil price. Comparative simulation results suggested that the proposed GA–NN approach is better than the baseline algorithms in terms of prediction accuracy and computational efficiency. Mann–Whitney test results indicated that the WTI crude oil price predicted by the proposed GA–NN and the observed price are statistically equal. Further comparison of the proposed GA–NN with previous studies indicated performance improvement over existing results. The proposed model can be useful in the formulation of policies related to international crude oil price estimations, development plans and industrial production.

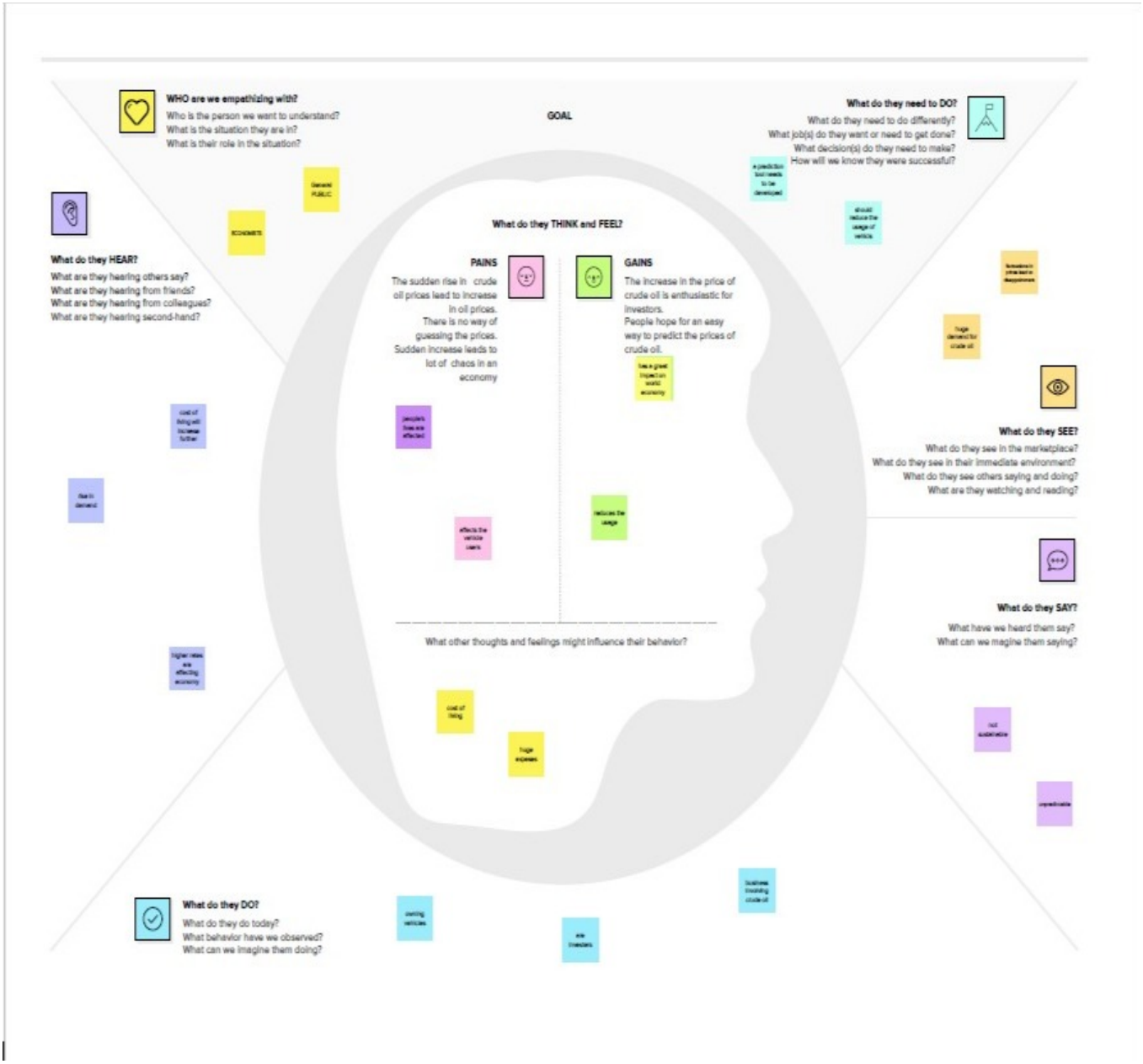
S. N. Abdullah and X. Zeng, Machine learning approach for crude oil price prediction with Artificial Neural Networks-Quantitative (ANN-Q) model:

Has observed and proposed,
The volatility of crude oil market and its chain effects to the world economy augmented the interest and fear of individuals, public and private sectors. Previous statistical and econometric techniques used for prediction, offer good results when dealing with linear data. Nevertheless, crude oil price series deal with high nonlinearity and irregular events. The continuous usage of statistical and econometric techniques for crude oil price prediction might demonstrate demotions to the prediction performance. Machine Learning and Computational Intelligence approach through combination of historical quantitative data with qualitative data from experts' view and news is a remedy proposed to predict this. This paper will discuss the first part of the research, focusing on to (i) the development of Hierarchical Conceptual (HC) model and (ii) the development of Artificial Neural Networks-Quantitative (ANN-Q) model.

3 IDEATION AND PROPOSED SOLUTION

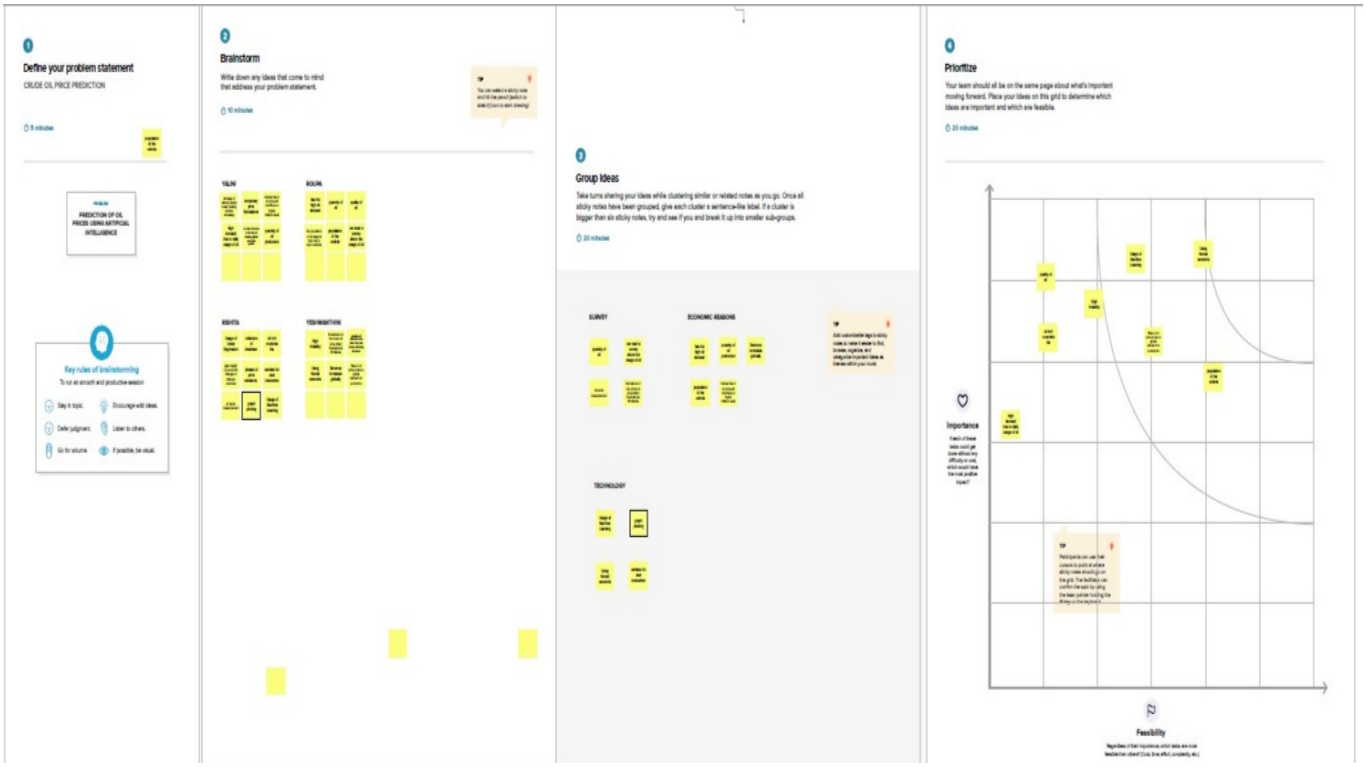
3.1 EMPATHY MAP CANVAS

It is a collaborative tool for gaining deeper insight about the target audience and it can represent a group of users.



3.2 IDEATION AND BRAINSTORMING

Ideation is often closely related to the practice of brainstorming, a specific technique that is utilized to generate new ideas. A principal difference between ideation and brainstorming is that ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity.



4 PROJECT DESIGN

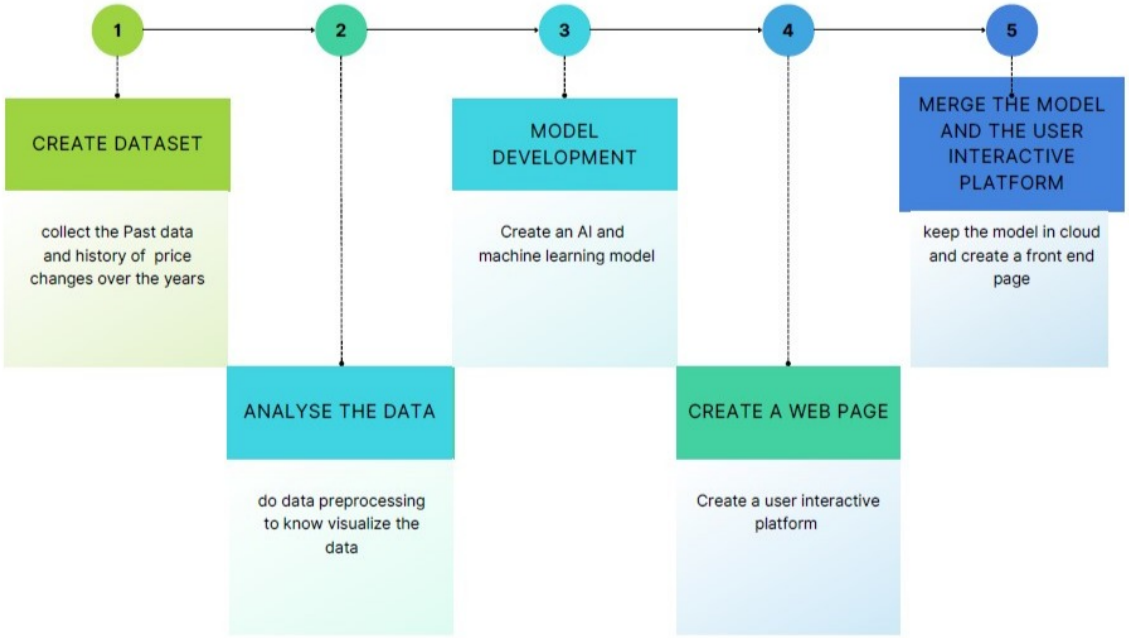
4.1 PROBLEM STATEMENT

4.2 PROBLEM SOLUTION FIT

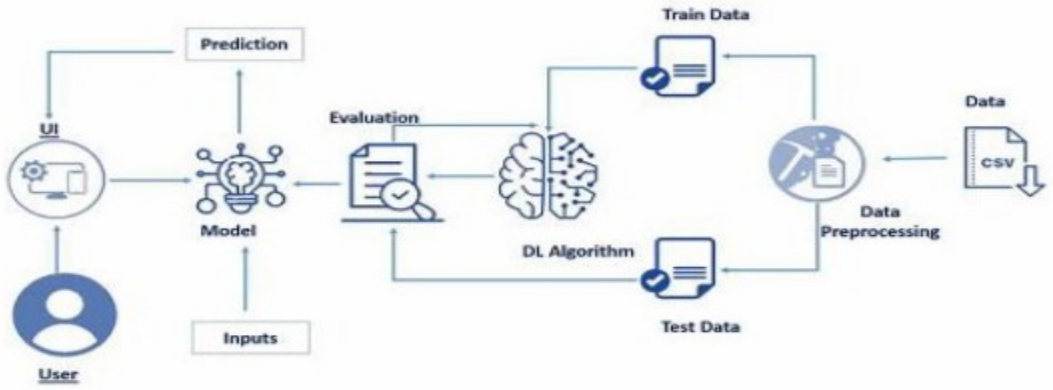
The design plan includes definition of problem statement, Idea of soluton, Social impact, Business model and scalabiity of solution.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The crude oil prices have been increasing tremendously in the recent times.
2.	Idea / Solution description	Creating a machine learning model and building a user interactive web age to show the predicted results.
3.	Novelty / Uniqueness	Usage of neural networks and deep learning
4.	Social Impact / Customer Satisfaction	Airlines, footwear, lubricants industry have lost the productivity too.
5.	Business Model (Revenue Model)	Crude oil price DespDev
6.	Scalability of the Solution	Can be used for prediction of other raw materials

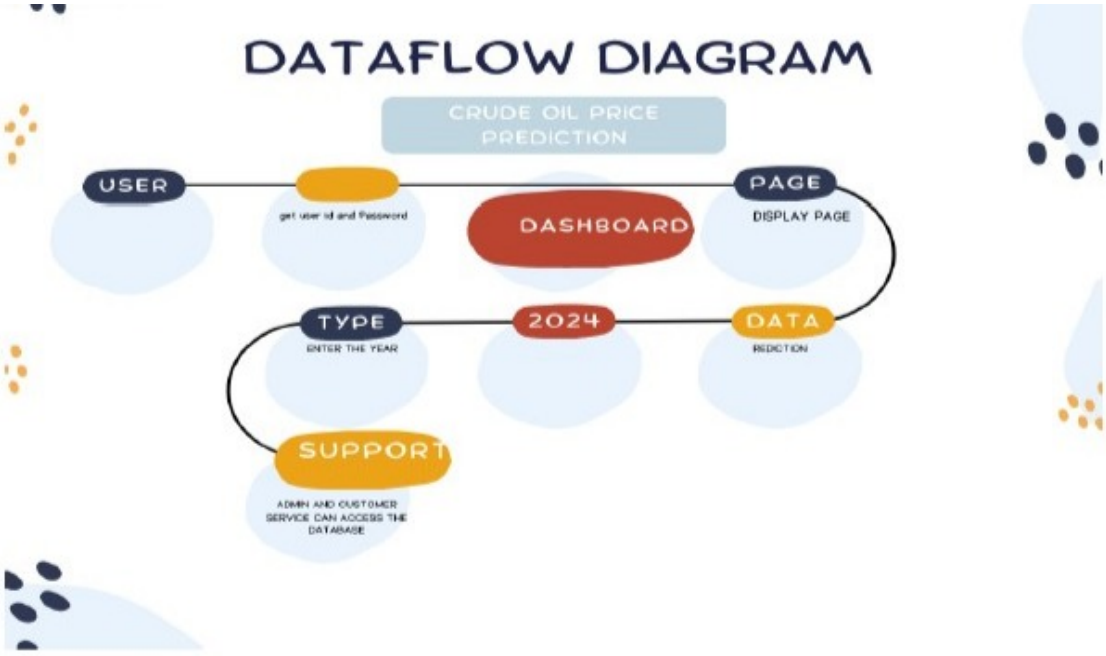
4.3 SOLUTION ARCHITECTURE



4.4 TECHNICAL ARCHITECTURE



4.5 DATA FLOW DIAGRAM



5 PROJECT PLANNING AND SCHEDULING

5.1 USER STORIES

User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can access using gmail account and received verification mail	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	Received verification mail	High	Sprint-3
Customer (Web user)	login	USN 6	I was able to access the web Portal through mail	Accessed my dashboard	high	Sprint 3
Customer Care Executive	login	USN 7	I was enabled access for the database of user and rectified the Problems	Accessed the user Portal for difficulty solving	high	Sprint 3
Administrator	admin	USN 8	Looked at user needs and responded	Manage the accounts Proerly	medium	Sprint 3

5.2 TECHNOLOGY STACK

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	WEB UI	HTML, CSS, FLASK
2.	Application Logic-1	COLLECT DATABASE	Python
3.	Application Logic-2	Save database	MySQL
4.	Application Logic-3	model	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	DL Algorithm	To redict
9.	Machine Learning Model	Purpose of Machine Learning Model	Prediction Model
10.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :IBM	Local, IBM Cloud, Kubernetes.

Table-2: Application Characteristics:

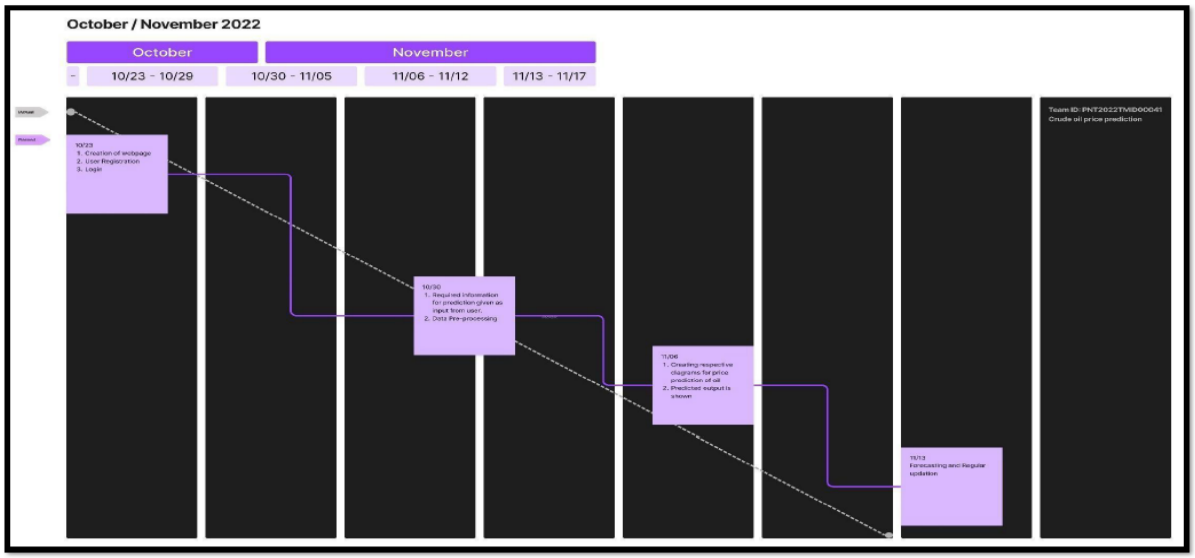
S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	IBM cloud	cloud
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	SHA-256, Encryptions
3.	Scalable Architecture	3 – tier, Micro-services	Machine learning
4.	Availability	distributed servers	FLASK
5.	Performance	15 REQUSTS PER SECOND	DL

6 SPRINT DELIVERY PLAN

6.1 SPRINT DELIVERY

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	10	High	Rishita.A
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	10	High	RISHITA A
Sprint-1	Login	USN-3	As a user, I can log into the application by entering email & password.	15	High	YESHWANTHINI
Sprint-2	Input Necessary Details	USN-4	As a user, I can give Input Details to Predict Likeliness of crude oil	15	High	YALINI C
Sprint-2	Data Pre-processing	USN-5	Transform raw data into suitable format for prediction.	15	High	YESHWANTHINI
Sprint-3	Prediction of Crude Oil Price	USN-6	As a user, I can predict Crude oil using machine learning model.	20	High	ROUPA DHARSHINI
Sprint-3		USN-7	As a user, I can get accurate prediction of crude oil	5	Medium	ROUPA DHARSHIN
Sprint-4	Review	USN-8	As a user, I can give feedback of the application.	20	High	YANINI C

6.2 BURNDOWN CHART



7 TECHNICAL REQUIREMENTS AND SOLUTIONING

- **ANACONDA ENVIRONMENT**
- **JUPYTER NOTEBOOK**
- **FLASK**

7.1 DATASET

We have downloaded the dataset from kaggle:
https://www.kaggle.com/rockbottom73/crude-oil-prices

7.2 DATA PREPROCESSING

Data pre processing includes the following tasks:

- Importing the libraries:

The required libraries to import the python scripts are,

7.2.1 NUMPY

Its the open source numerical python script.It contains the multi dimensional array and matrix data structure also perform the mathematical operation on array such as trigonometric,statistical and algebra.

7.2.2 PANDAS

One of the top python programming languages and it is fast , flexible and easy to use the open source data analysis.

7.2.3 MATPLOTLIB

Creating static, animated and interactive visualizing of python

- Importing the dataset:

➤ You might have your data in csv files, xlsx files

➤ Loading the excel sheet into pandas

➤ Locate the dictionary into the excel sheet to keep the data set more efficient .

➤ If the data set in some other location, we can found out with the help of pandas.

➤ Loading the pandas into same dictionary so we can read directly and easy.

7.2.4 HANDLING MISSING DATA

➤ After loading the dataset check the rows and columns for their null values with complete information.

➤ If there is any null values ,following can be done:

➤ Using data imputation the data is imputing in sklearn.

➤ Filling the NaN values with help of median, mode and mean using fillna() method.

➤ Delete the records.

➤ Now we can see the null values in the closing value column and also check how many numbers of null values in the column using sum() function.

- Drop the null from the column.
- Axis=0 drop the row.
- Data frame has to change permanent indicates, ' in place=True'.
- Reset_index consider the closing value column to reset the index of data frame list of integer from 0 to length of data.

7.2.5 FEATURE SCALING

Feature scaling to normalize the independent variables to scale the crude oil prices between (0 ,1) to avoid the computation . Some of the common methods are standardization and Normalization.

7.2.6 STANDARDIZATION

The process of developing and implementing the technical products and stability of the products.

7.2.7 NORMALIZATION

The process of organizing the data base or data frame to reduce the redundancy and improve the integrity of the data , improve and simplifies the data design.

7.2.8 DATA VISUALIZATION

- In data visualization the data set were presented in graphical pattern, it helps to detects the pattern, levels, correlation and trends in recent days which has not been detected in text format.

- To understand the data algorithm and relationship even the well performed machine can also perform poorly which wasn't visualize properly and understood. We need libraries like Matplotlip and sea born to visualize the data set.

Matplotlip is the 2Dpython libraries which helps to plots, bar charts etc..,

Using the matplotlip and sea born libraries are:

Basic properties of plotting the graphs are:

- **X label:** set the x- axis.
- **Y label:**set the y-axis.
- **Title:** set the title for the axis.
- **Legend:**place the legend for the axes.

There are different types of analysis ;

UNIVARIATE: Using the single features to analyses the properties.

BIVARIATE: Compare the data between the 2 features.

MULTIVARIATE: Comparing more than 2 variables.

7.3 SPLITTING DATA INTO TRAIN AND TEST

On working on model have to test and train the model and have to set the data frame . After training we have to test the some dataset.Train and test the dataset in

the different set of data frames in the earlier ways. During the development phase there will be always the lost of dataset. So that the dataset have to split into two different set one is to training and other one is for testing.

For the time series data, and the data sequence are important. The size of train and test the data after splitting.

7.4 CREATING A DATASET WITH SLIDING WINDOW

A special type of data set or structure needed to cover n-type stamp.LSTM will predict the n+1 price.The number of data stamp is set of 10.

The function has two arguments dataset which is Numpy want to convert the dataset and the time_step will predict the previous time set of the input variables, and defaulted to 1.

The default will create the dataset with X which will predict the price of crude oil at the time t, and Y will predict the crude oil at the next time (n+t).

For LSTM model it requires the train and test of X in three dimensional array for building the model.

7.5 MODEL BUILDING

Model building includes,

1. Model building blocks importation.
2. Intialise the model

3. Add the LSTM layers
4. Add output layer
5. Learning process.
6. Training the model
7. Model evaluation.
8. Save the model
9. Test the model.

7.5.1 IMPORT THE MODEL BULIDING LIBRARIES

Importing the necessary model libraries.

7.5.2 INTIALISE THE MODEL

Linear stack of layer is the sequential model.Create the sequential order by passing the list of layer. Model imports the sequential order in keras.

7.5.3 ADDING THE LSTM LAYER

Units is the number of LSTM neurons layer. Fifty neurons will give the model high dimensionality, enough to capture the upwards and downward trends.

Return_sequences is True as we need to add another LSTM layer after the current one. input_shape corresponds to the number of time stamps and the number of indicators.

7.5.4 ADDING OUTPUT LAYER

Dense layer is a deeply connected neural network layer. It is the most common and frequently used layer.

The output dimension is one since we are predicting one price each time.

Understanding the model is a crucial phase to use it for training properly and prediction purposes.

Keras provides a simple method, summary to get the full information about the model and its layers.

7.5.5 CONFIGURE THE LEARNING PROCESS

Compilation is the final step in creating a model. Once the compilation is done, we can move on to the training phase

The loss function is used to find error or deviation in the learning process. Keras requires loss function during the model compilation process.

Optimization is an important process that optimizes the input weights by comparing the prediction and the loss function. Here we are using adam optimizer

Metrics are used to evaluate the performance of your model. It is similar to loss function, but not used in the training process

7.5.6 TRAIN THE MODEL

RNN weights are updated every 64 stock prices with a batch size of 64. Try more batches and epochs if the loss of the model is not converging.

Arguments:

Epochs: an integer and number of epochs we want to train our model for.
An inputs and targets list
A generator
An inputs, targets, and sample_weights list which can be used to evaluate
The loss and metrics for any model after any epoch has ended.

7.6 MODEL EVALUATION

we need to check to see how well our model is performing on the test data.

Regression Evaluation Metrics

7.6.1 Mean Squared Error (MSE)

MSE or Mean Squared Error is one of the most preferred metrics for regression problems.

It is simply the average of the squared difference between the target value and the value predicted by the regression model.

As it squares the differences, it penalizes even a small error which leads to over-estimation of how bad the model is.

It is preferred more than other metrics because it is differentiable and hence can be optimized better.

7.6.2 RMSE:Root Mean Square Error

RMSE is the square root of the averaged squared difference between the target value and the value predicted by the model.

It is preferred more in some cases because the errors are first squared before averaging which poses a high penalty on large errors.

This implies that RMSE is useful when large errors are undesired.

7.7 SAVE THE MODEL

An H5 file is a data file saved in the Hierarchical Data Format (HDF). It contains multidimensional arrays of scientific data.

7.7.1 TEST THE MODEL

Shift the predictions so that they align on the x-axis with the original dataset.

Once prepared, the data is plotted, showing the original dataset in blue, the predictions for the training dataset in green, the predictions on the unseen test dataset in orange.

8 APPLICATION BUILDING:

A UI is provided for the users where he has to enter the values for predictions.

The entered values are given to the saved model and prediction is showcased on the UI.

8.1 CREATE AN HTML FILE

We use HTML to create the front end part of the web page.
Here, we created 2 HTML pages- index.html, web.html.
index.html displays the home page.
web.html accepts the values from the input and displays the prediction.

8.2 INDEX PAGE

```
<!DOCTYPE html>
<html lang="en" >
<head>
  <meta charset="UTF-8">
  <title>Sign Up Signin Form Template Example</title>
  <link rel="stylesheet" href="./style.css">

</head>
<body>
<!-- partial:index.partial.html -->
<html lang="en">
<head>
  <!-- Latest compiled and minified CSS -->
<link                                                                    rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">
  </head>

<body>
<div id="form">
  <div class="container">
     <div class="col-lg-6 col-lg-offset-3 col-md-6 col-md-offset-3 col-md-8 col-md-offset-
2">
```

```
<div id="userform">
 <ul class="nav nav-tabs nav-justified" role="tablist">
  <li class="active"><a href="#signup"  role="tab" data-toggle="tab">Sign up</a></li>
  <li><a href="#login"  role="tab" data-toggle="tab">Log in</a></li>
 </ul>
 <div class="tab-content">
  <div class="tab-pane fade active in" id="signup">
   <h2 class="text-uppercase text-center"> Sign Up for Free</h2>
   <form id="signup">
    <div class="row">
     <div class="col-xs-12 col-sm-6">
      <div class="form-group">
       <label>First Name<span class="req">*</span> </label>
            <input type="text" class="form-control" id="first_name" required data-validation-required-message="Please enter your name." autocomplete="off">
       <p class="help-block text-danger"></p>
      </div>
     </div>
     <div class="col-xs-12 col-sm-6">
      <div class="form-group">
       <label> Last Name<span class="req">*</span> </label>
            <input type="text" class="form-control" id="last_name" required data-validation-required-message="Please enter your name." autocomplete="off">
       <p class="help-block text-danger"></p>
      </div>
     </div>
    </div>
    <div class="form-group">
     <label> Your Email<span class="req">*</span> </label>
        <input type="email" class="form-control" id="email" required data-validation-required-message="Please enter your email address." autocomplete="off">
     <p class="help-block text-danger"></p>
    </div>
    <div class="form-group">
     <label> Your Phone<span class="req">*</span> </label>
         <input type="tel" class="form-control" id="phone" required data-validation-required-message="Please enter your phone number." autocomplete="off">
```

```
            <p class="help-block text-danger"></p>
           </div>
           <div class="form-group">
            <label> Password<span class="req">*</span> </label>
                  <input type="password" class="form-control" id="password" required data-
validation-required-message="Please enter your password" autocomplete="off">
            <p class="help-block text-danger"></p>
           </div>
           <div class="mrgn-30-top">
            <button type="submit" class="btn btn-larger btn-block"/>
            Sign up
            </button>
           </div>
          </form>
        </div>
        <div class="tab-pane fade in" id="login">
         <h2 class="text-uppercase text-center"> Log in</h2>
         <form id="login">
          <div class="form-group">
           <label> Your Email<span class="req">*</span> </label>
              <input type="email" class="form-control" id="email" required data-validation-
required-message="Please enter your email address." autocomplete="off">
           <p class="help-block text-danger"></p>
          </div>
          <div class="form-group">
           <label> Password<span class="req">*</span> </label>
                 <input type="password" class="form-control" id="password" required data-
validation-required-message="Please enter your password" autocomplete="off">
           <p class="help-block text-danger"></p>
          </div>
          <div class="mrgn-30-top">
           <button type="submit" class="btn btn-larger btn-block"/>
           Log in
           </button>
          </div>
         </form>
        </div>
```

```
      </div>
     </div>
    </div>
  </div>
  <!-- /.container -->
</div>
<script src="//code.jquery.com/jquery-1.11.3.min.js"></script>
<!-- Latest compiled and minified JavaScript -->
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></script>
</body>
<!-- partial -->
  <script  src="./script.js"></script>

</body>
</html>
```

8.3 WEB PAGE

```
<html>

<style>
div.header{
  top: 0;
  position: fixed;
  padding-left: 400px;}
div.header1{
  top:20;
  position: fixed;
  padding-left: 490px;
}

*{
  margin:0;
          padding:0;
          border:0;
          outline:0;
          text-decoration:none;
```

```
        font-family:montserrat;
}

body
{
background-image:url('{{url_for('static', background="oil-2_0-sixteen_nine.jpg")}}');
background-position: center top;
font-family:sans-serif;
background-size:cover;
margin-top:40px;
}

.main input[type="text"],.main input[type="text"],.main input[type="text"],.main
input[type="text"],.main input[type="text"],.main input[type="text"],.main
input[type="text"]{
        border:0;
        background:none;
        display:block;
        margin:20px auto;
        text-align:center;
        border:2px solid black;
        padding:15px 3px;
        width:400px;
        outline:none;
        color:black;
        border-radius:0px;
        transition:0.25s;
        font-size:20;

}
.bor{
border:0;
        background:none;
        display:block;
        margin:20px auto;
        text-align:center;
        border:2px solid black;
```

```
        padding:10px 3px;
        width:500px;
        outline:none;
        color:black;
        transition:0.25s;}
.main input[type="text"]:focus,.main input[type="text"]:focus,.main input[type="text"]:focus,.main input[type="text"]:focus,.main input[type="text"]:focus,.main input[type="text"]:focus,.main input[type="text"]:focus{
        width:280px;
        color: black;
        border-color:black;
}
.logbtn{
        display:block;
        width:35%;
        height:50px;
        border:none;
        border-radius:24px;
        background:linear-gradient(120deg,#3498db,#8e44ad,#3498db,#8e44ad);
        background-size:200%;
        color:black;
        outline:none;
        cursor:pointer;
        transition:.5s;
        font-size:25;
}
.logbtn:hover{
        background-center;
}

input::placeholder{
        color:purple;
        font-family: verdana;
        font-size: 15;
}
.bottom-text{
        margin-top:60px;
```

```
        text-align:center;
        font-size:13px;

}

</style>
<body>
<div class="navbar">
<a href="/" style="font-size: 20px;color: blue;font-family: verdana;" >Contact | </a>
<a href="/about" style="font-size: 20px;color: blue;font-family: verdana;" >Home</a>
<br>
</div>
        <center><h1 style="font-family:verdana;">CRUDE OIL PRICE
PREDICTOR</h1></center>


<br><br><br><br>
        <form style="color: black;" class="main" action="/login" method="post">
                <br>
                <font size=20><input type="text" name="year1" placeholder="Enter
previous 10th day price" style="color:black;"/></font>
                <font size=20><input type="text" name="year2" placeholder="Enter
previous 9th day price"/></font>
                <font size=20><input type="text" name="year3" placeholder="Enter
previous 8th day price"/></font>
                <font size=20><input type="text" name="year4" placeholder="Enter
previous 7th day price"/></font>
                <font size=20><input type="text" name="year5" placeholder="Enter
previous 6th day price"/></font>
                <font size=20><input type="text" name="year6" placeholder="Enter
previous 5th day price"/></font>
                <font size=20><input type="text" name="year7" placeholder="Enter
previous 4th day price"/></font>
                <font size=20><input type="text" name="year8" placeholder="Enter
previous 3th day price"/></font>
                <font size=20><input type="text" name="year9" placeholder="Enter
previous 2nd day price"/></font>
```

```
                <font size=20><input type="text" name="year10" placeholder="Enter previous 1st day price"/></font>
                <center><input type="submit" class="logbtn" value="Predict"></center>
                <div class="bor"><b><font color="white" size=5>{{showcase}}</font></b></div>
        </form>
</div>
</body>

</html>
```

9 BUILD PYTHON CODE

We have a built a flask file 'app.py' which is a web framework written in python for server-side scripting.

- Let's see step by step procedure for building the backend application.
- The app starts running when the "__name__" constructor is called in main.
- render_template is used to return HTML files.
- "GET" method is used to take input from the user.
- "POST" method is used to display the output to the user.
- Importing Libraries
- Routing to the html Page
- For predicting next day's crude oil prices we consider n_steps=10.
- the input for prediction, index starting from the date 10 days before the first date in the test dataset.
- Then, reshape the inputs to have only 1 column and predict using model_predict predefined function.

```
import numpy as np  # used for numerical analysis
from flask import Flask, render_template, request, url_for, redirect  # Flask is a application used to run/serve our application
# request is used to access the file which is uploaded by the user in our application
# render_template is used for rendering the html pages
from tensorflow.keras.models import load_model  # we are loading our model from keras


app = Flask(__name__)  # our flask app
model = load_model('crude_oil_price_prediction.h5')  # loading the model in the flask app

@app.route('/', methods=['GET', 'POST'])
def home():
    error = None
    if request.method == 'POST':
        if request.form['username'] != "PNT2022TMID43400" or request.form['password'] != "7155":
            error = 'Invalid Credentials. Please try again.'
        else:
            return redirect(url_for('mains'))
    return render_template('login.html', error=error)


@app.route('/mains', methods=['GET', 'POST'])
```

```
def mains():
    return render_template('index.html')

@app.route('/stats', methods=['GET', 'POST'])
def stats():
    return render_template('stats.html')



@app.route('/about')
def home1():
    return render_template("index.html")  # rendering html template


@app.route('/predict')
def home2():
    return render_template("web.html")  # rendering html template

@app.route('/contact')
def contact():
    return render_template("contact.html")


@app.route('/login', methods=['POST'])  # route for our prediction
def login():
    a = request.form['year1']
    b = request.form['year2']
    c = request.form['year3']
    d = request.form['year4']
    e = request.form['year5']
    f = request.form['year6']
    g = request.form['year7']
    h = request.form['year8']
    i = request.form['year9']
    j = request.form['year10']  # requesting the file
    x_input = [[float(a), float(b), float(c), float(d), float(e), float(f), float(g), float(h), float(i), float(j)]]
    print(x_input)
    lst_output = model.predict(x_input)
    lst_output = np.round(lst_output[0][0], 2)
    return render_template("web.html", showcase='The Predicted crude oil price is : Rs. '+str(lst_output))


if __name__ == '__main__':
    app.run(debug=False)
```

9.1 RUN THE APP IN LOCAL BROWSER

- Open anaconda prompt from the start menu
- Navigate to the folder where your python script is.
- Now type “python app.py” command
- Navigate to the localhost where you can view your webpage

9.2 SHOWCASTING PREDICTION ON UI

- In home page we get the summary of the project.
- For the time series approach the user has to give the past 10 days price of crude oil to predict the future crude oil prediction.
- If we give input of past 10 days and click the predict for the next day.
- The output will be displayed to the user interface.

9.3 TRAIN THE MODEL ON IBM

Build a Machine Learning Model and deploy it on the IBM Cloud.

9.3.1 REGISTER FOR IBM CLOUD

create an IBM cloud account to login and also for train the model.

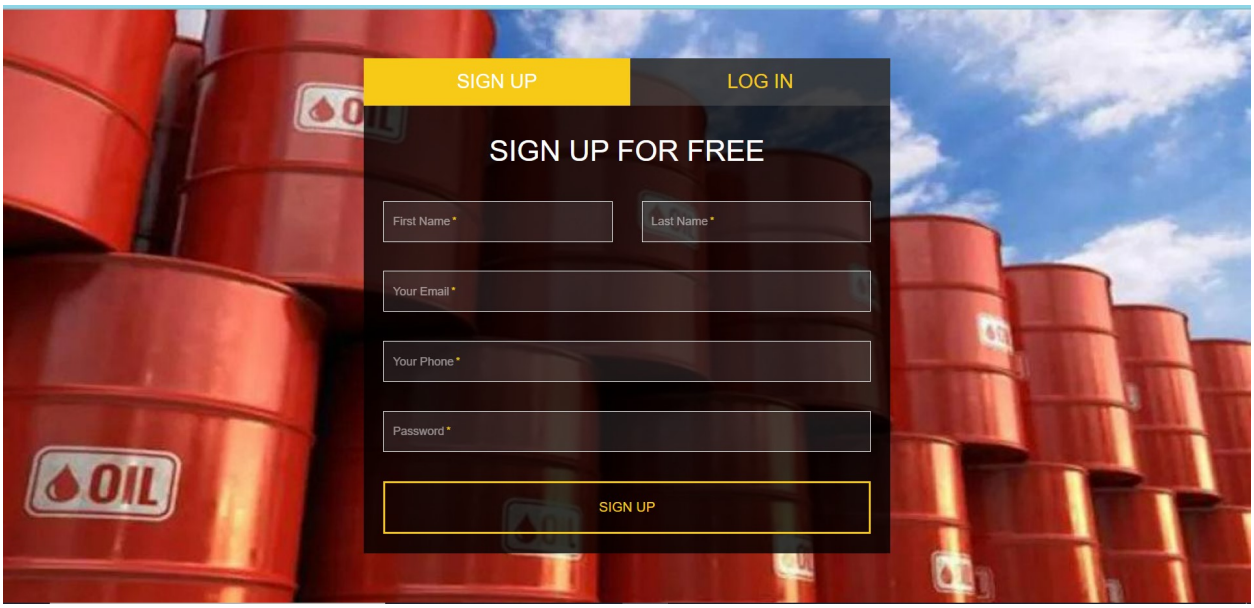
9.3.2 TRAIN THE ML MODEL ON IBM

Train the machine learning model on the IBM watson.

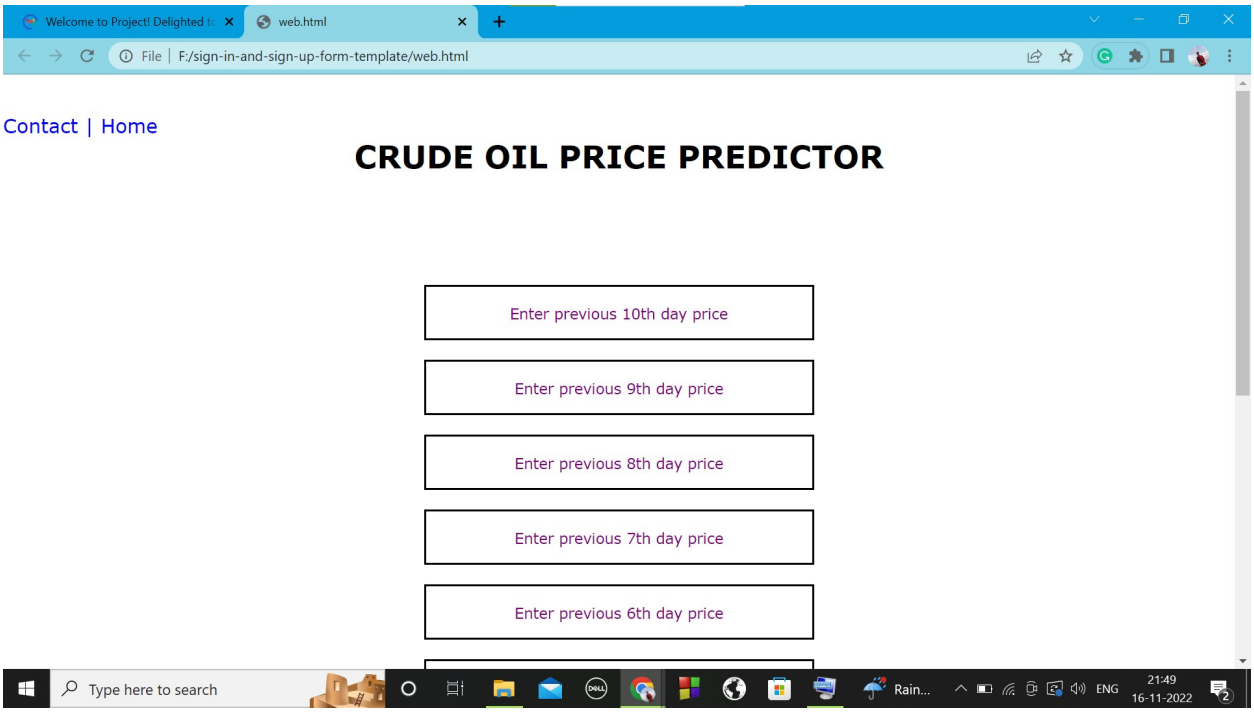
10 RESULT

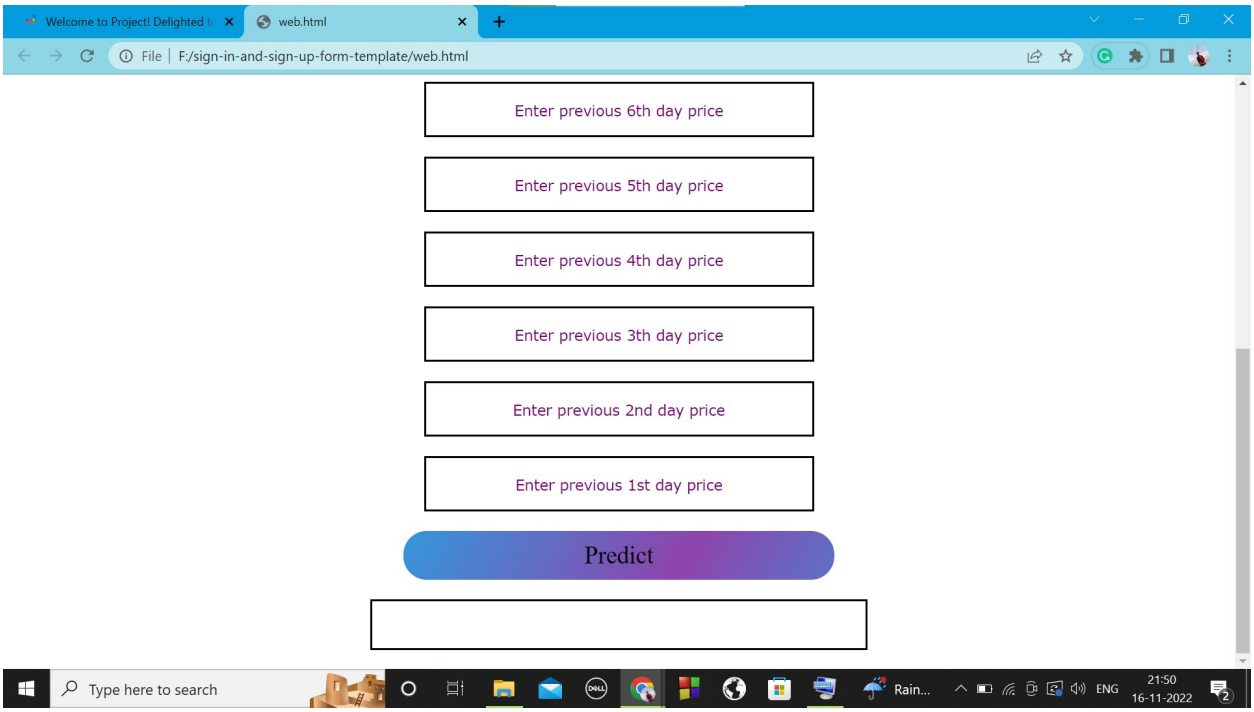
A Machine Learning model using LSTM was developed and a user interface was created.

10.1 USER INTERFACE



10.2 INDEX PAGE

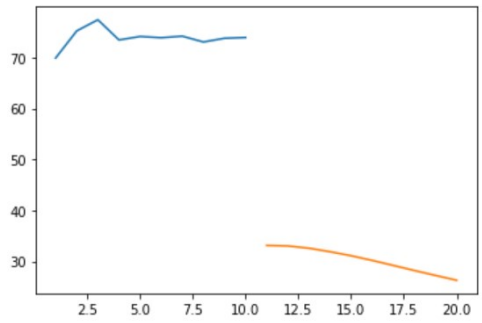




10.3 TESTING THE MODEL

```
plt.plot(day_new,scaler.inverse_transform(dataoil[8206:]))
plt.plot(day_pred,scaler.inverse_transform(lst_output))
```

```
[<matplotlib.lines.Line2D at 0x1b0e9be0760>]
```




```
df3=dataoil.tolist()
df3.extend(lst_output)
plt.plot(df3[8100:])
```

```
[<matplotlib.lines.Line2D at 0x1b0eaf28220>]
```