**Crypto price prediction using FbProphet.**

**Project Report**

1. **INTRODUCTION** 1.1 Project Overview 1.2 Purpose

1. **LITERATURE SURVEY**2.1 Existing problem  
   2.2 References  
   2.3 Problem Statement Definition
2. **IDEATION & PROPOSED SOLUTION**

3.1 Empathy Map Canvas  
3.2 Ideation & Brainstorming

1. **REQUIREMENT ANALYSIS**

4.1 Functional requirement

4.2 Non-Functional requirements

1. **PROJECT DESIGN**

**5.1 Data Flow Diagrams & User Stories**

**5.2 Solution Architecture**

1. **PROJECT PLANNING & SCHEDULING**

**6.1 Technical Architecture  
6.2 Sprint Planning & Estimation 6.3 Sprint Delivery Schedule**

1. **CODING & SOLUTIONING (Explain the features added in the project along with code)**

**7.1 Feature 1  
7.2 Feature 2  
7.3 Database Schema (if Applicable)**

1. **PERFORMANCE TESTING**

**8.1 Performance Metrics**

1. **RESULTS**

**9.1 Output Screenshots**

1. **ADVANTAGES & DISADVANTAGES**
2. **CONCLUSION**
3. **FUTURE SCOPE**
4. **APPENDIX**

**Source Code  
GitHub & Project Demo Link**

1. **Project Introduction**

The core idea of this project is to be able to predict the price of Bitcoin which is the most valuable cryptocurrency in the world with reasonable accuracy, using time series forecasting libraries like FbProbhet.

This will not only enable us to take trades on the basis of future price but on a larger scale if the model is accurate enough it can also be used to make crypto market more predictable and hence more acceptable in general public who usually wants to avoid high risk environments.

Bitcoin prediction can be useful in various ways, offering valuable insights to investors, traders, and the broader cryptocurrency community. By analyzing historical data and market trends, these predictions can help individuals make informed investment decisions, manage risk, and optimize their portfolios. They can also serve as a foundation for trading strategies and automated algorithms, enabling users to capitalize on price movements.

Additionally, Bitcoin predictions can foster awareness, education, and market sentiment analysis, contributing to a better understanding of the cryptocurrency market's dynamics and its potential for both gains and losses. However, it's essential to remember that cryptocurrency markets are highly volatile, and predictions are never fool proof, so they should be used in conjunction with other research and risk management strategies.

1. **LITERATURE SURVEY**

**Time Series Forecasting in Finance:** Explore research papers and articles on the application of time series forecasting in financial markets. Understand different methods and algorithms used for predicting financial time series data.

**Cryptocurrency Price Prediction:** Look into studies specifically focused on predicting cryptocurrency prices. Analyse the challenges and methodologies employed in predicting the volatile nature of crypto assets.

**FbProphet and Time Series Forecasting:** Investigate academic papers or documentation related to FbProphet to understand its underlying principles. Examine case studies where FbProphet has been successfully applied in various domains.

**Web-based Machine Learning Model Deployment:** Explore literature on deploying machine learning models on the web, especially using Flask. Understand the best practices, challenges, and security considerations in deploying predictive models online.

**User Acceptance of Predictive Models:** Search for literature discussing user perceptions and acceptance of predictive models, especially in the context of financial markets. Consider ethical implications and transparency in deploying predictive models to a wider audience.

**Integration of Data Sources (Yahoo Finance**): Investigate how financial data from sources like Yahoo Finance is used in academic and industry research. Understand the challenges and best practices in collecting and pre-processing financial time series data.

**Evaluation Metrics for Predictive Models:** Explore literature on metrics used to evaluate the performance of time series forecasting models. Understand the limitations of common metrics and any proposed alternatives.

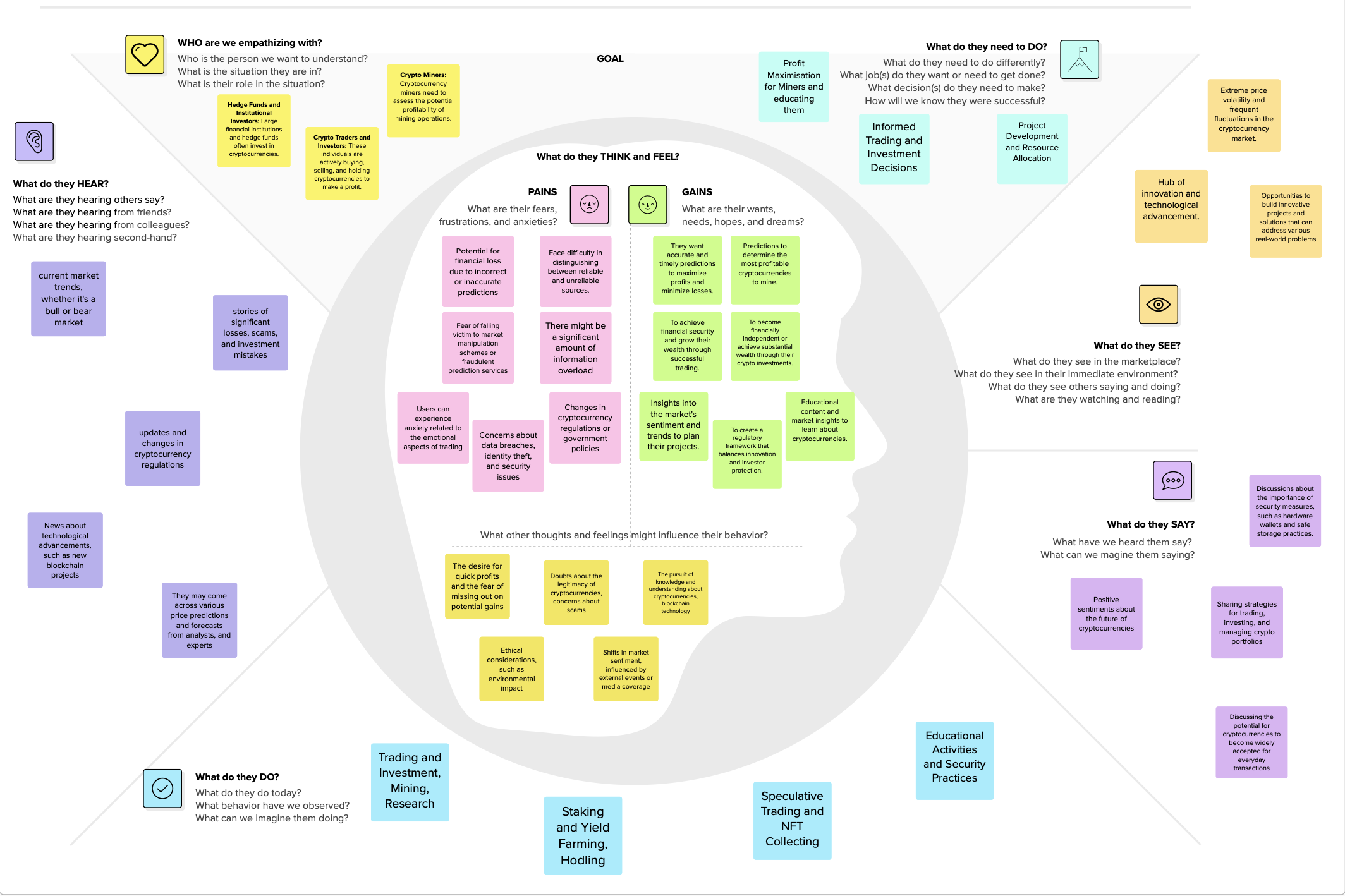
**Impact of Predictive Models on Financial Markets:** Research articles discussing the potential impact of accurate predictive models on financial markets. Consider both positive and negative implications and the overall market dynamics.

**Machine Learning Model Interpretability:** Look into research on making machine learning models more interpretable, especially in the context of financial predictions. Understand the importance of model interpretability for user trust and acceptance.

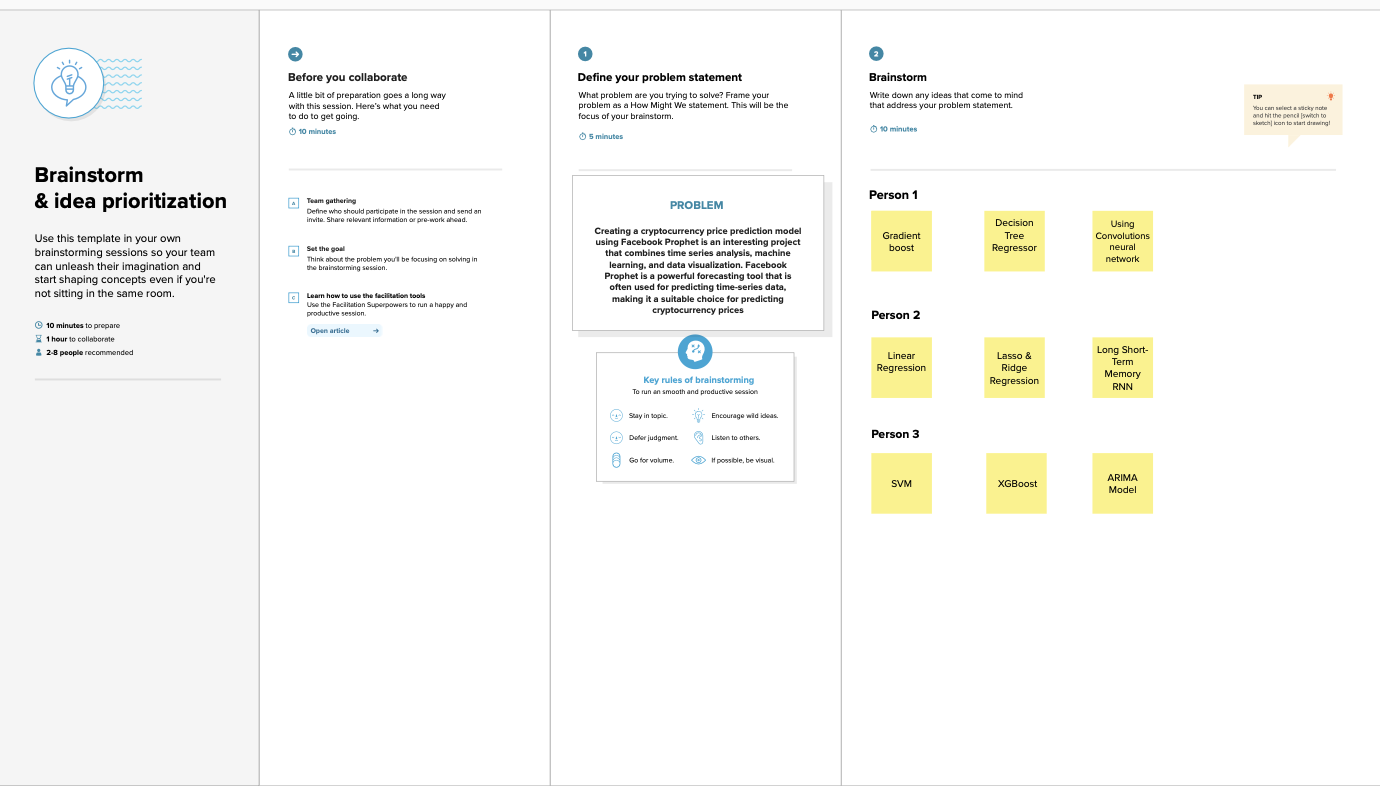
**Future Trends in Crypto and Finance Prediction:** Explore recent publications discussing emerging trends and advancements in the field of cryptocurrency prediction and financial forecasting.

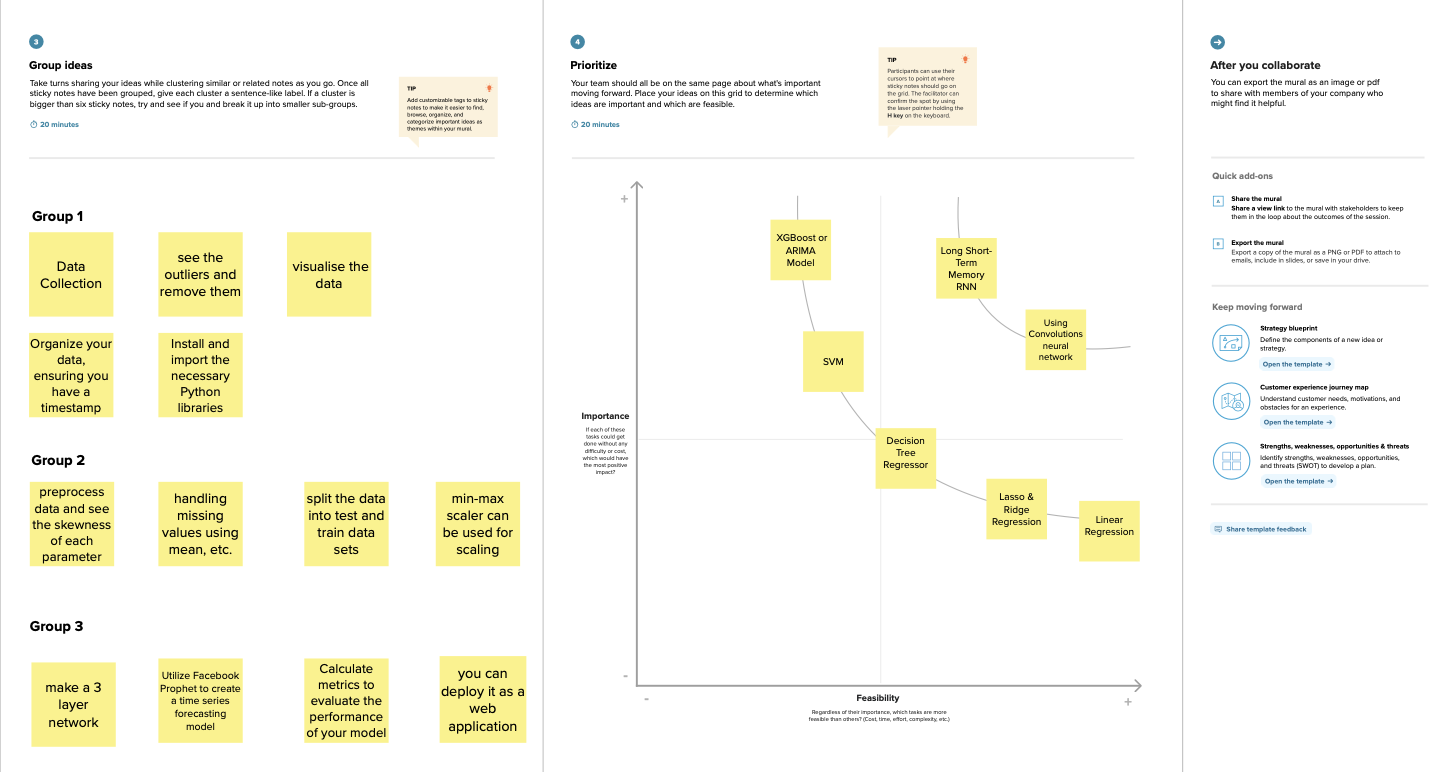
1. **IDEATION & PROPOSED SOLUTION**

3.1 Empathy Map

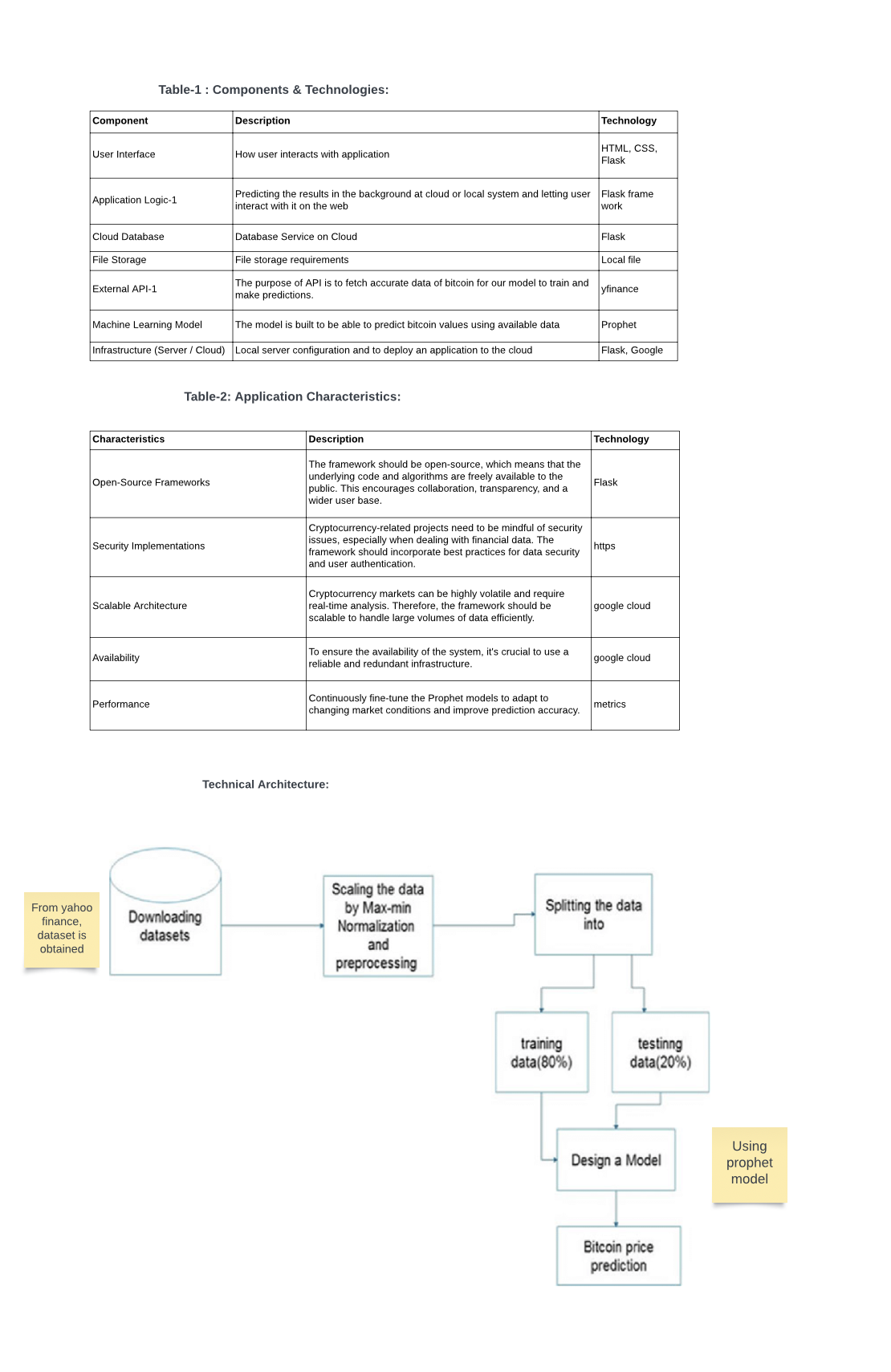


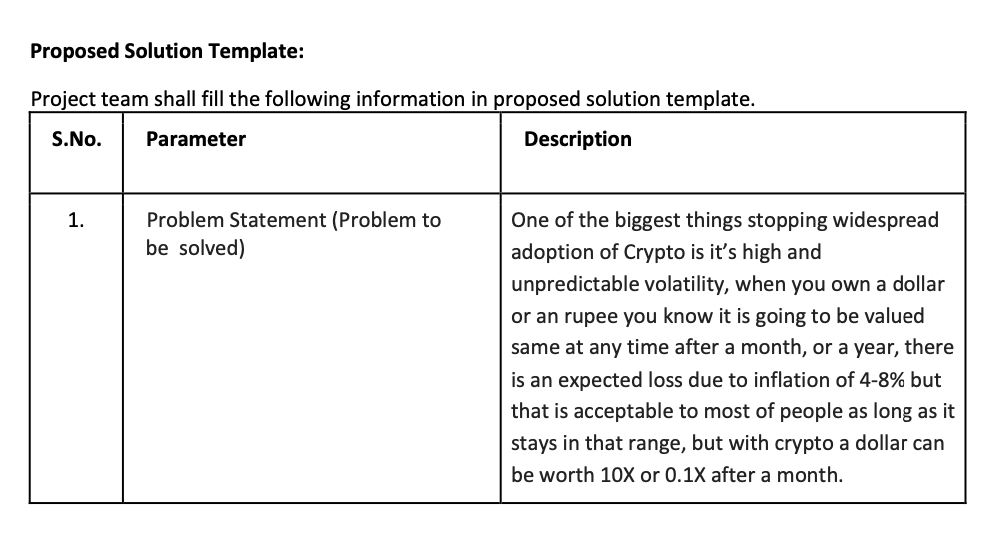
* 1. Brain Storming



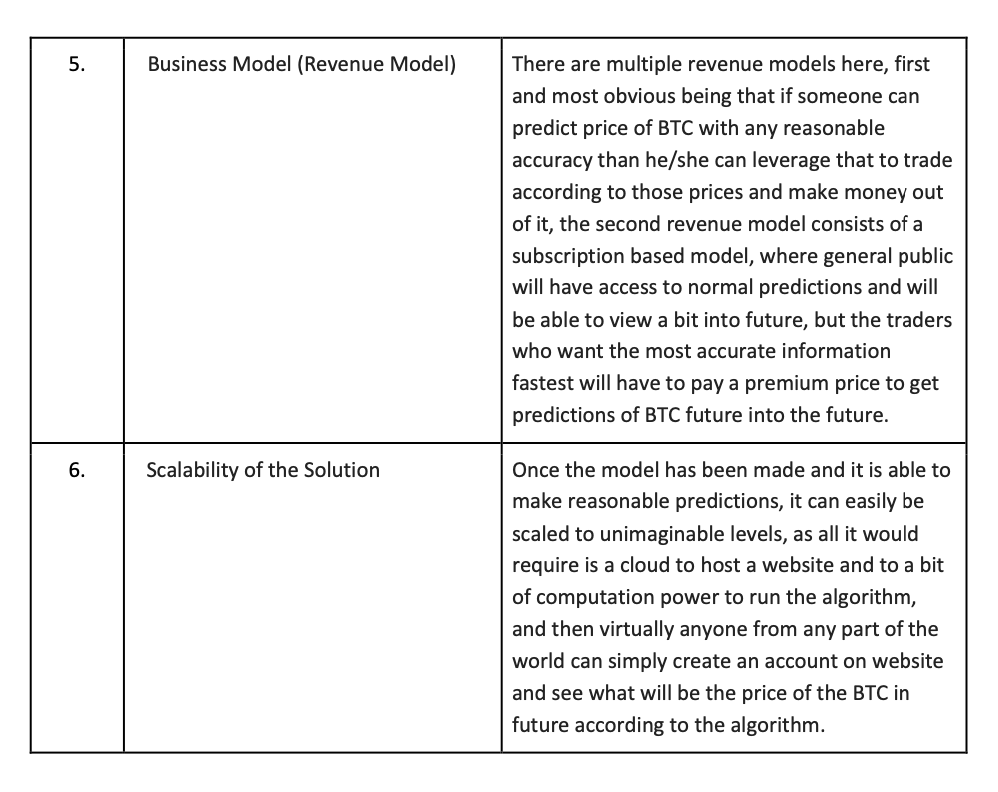


1. Requirement Analysis



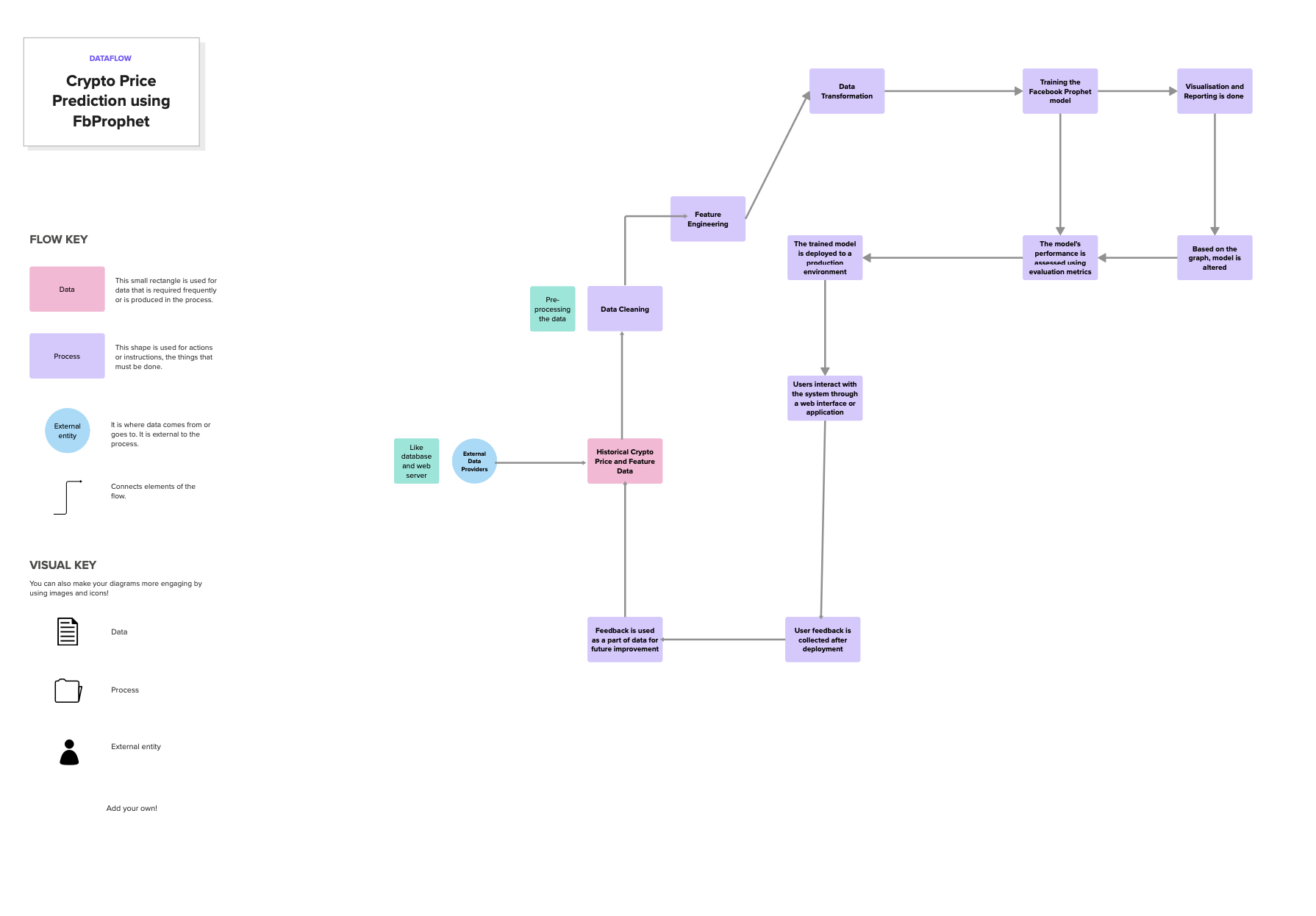




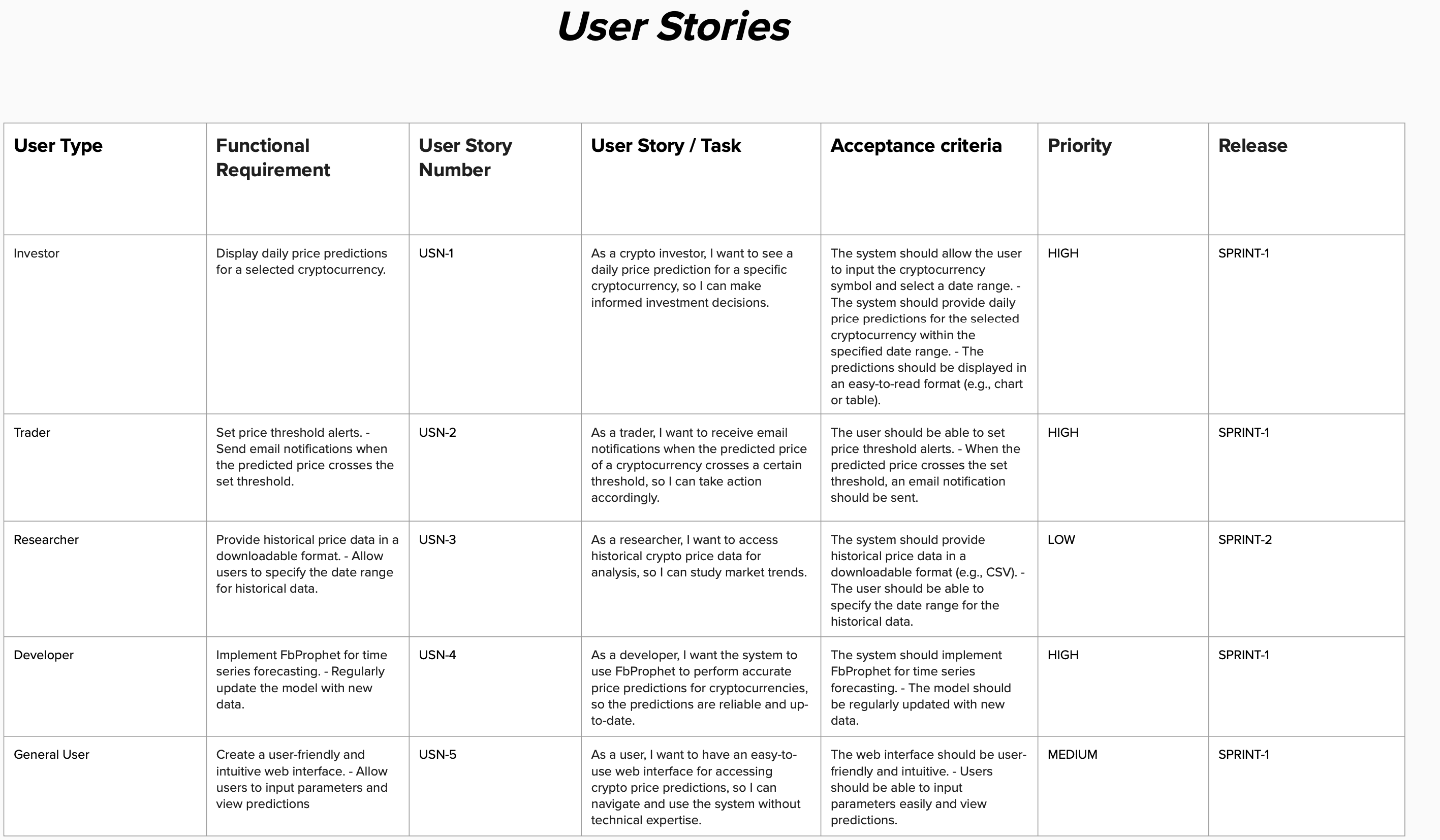


1. Project Design

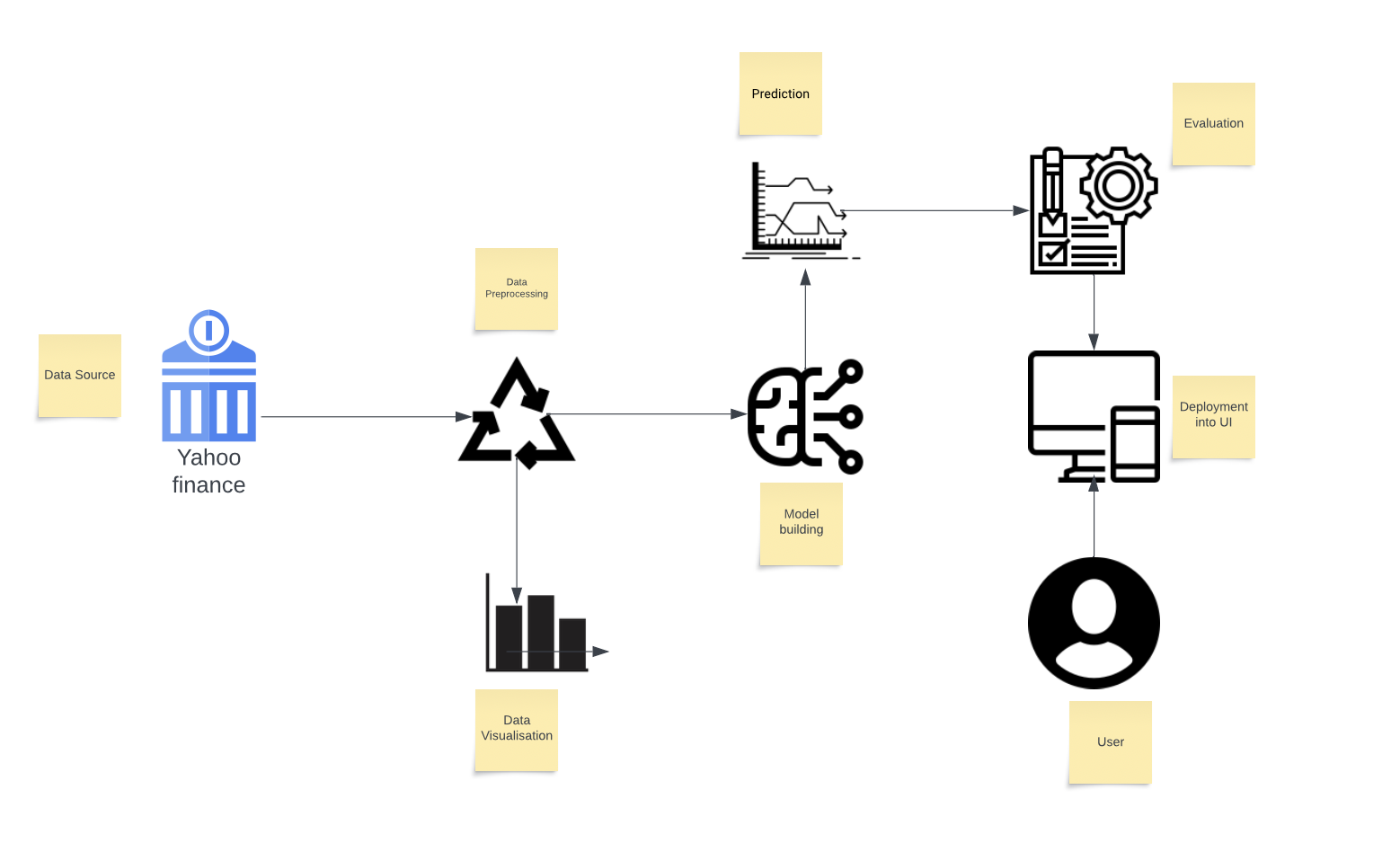
Data flow diagram:



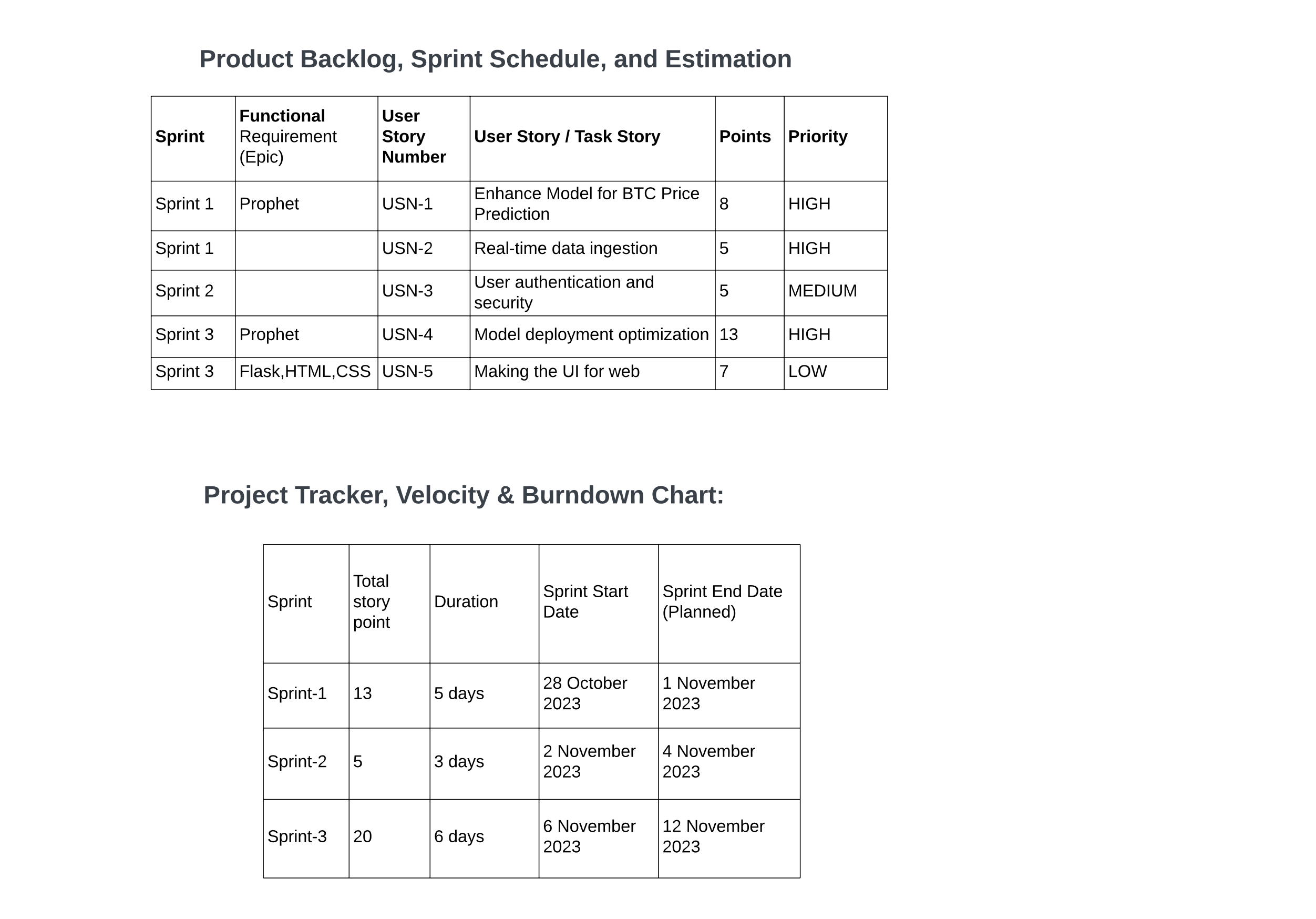
User stories:

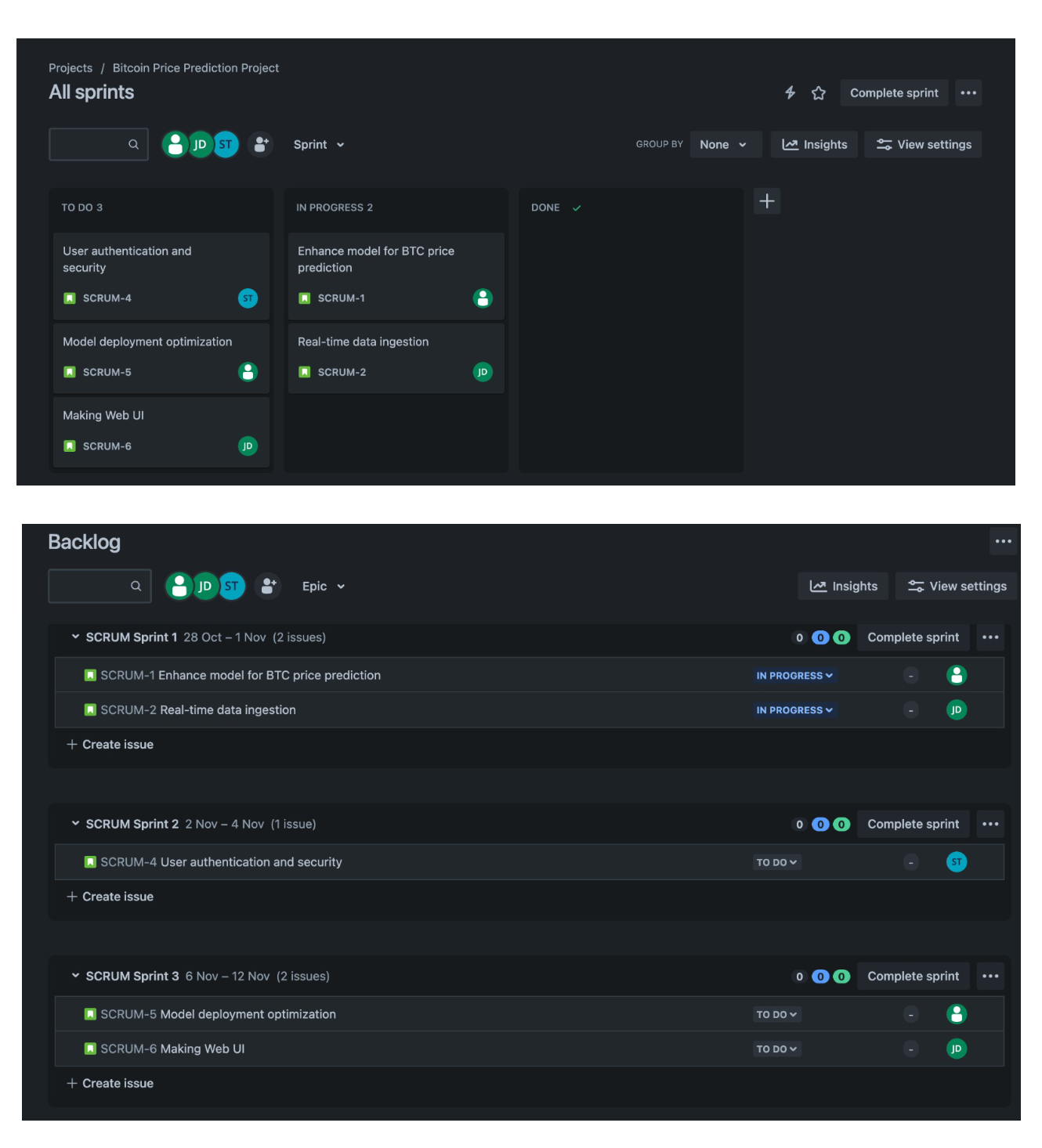


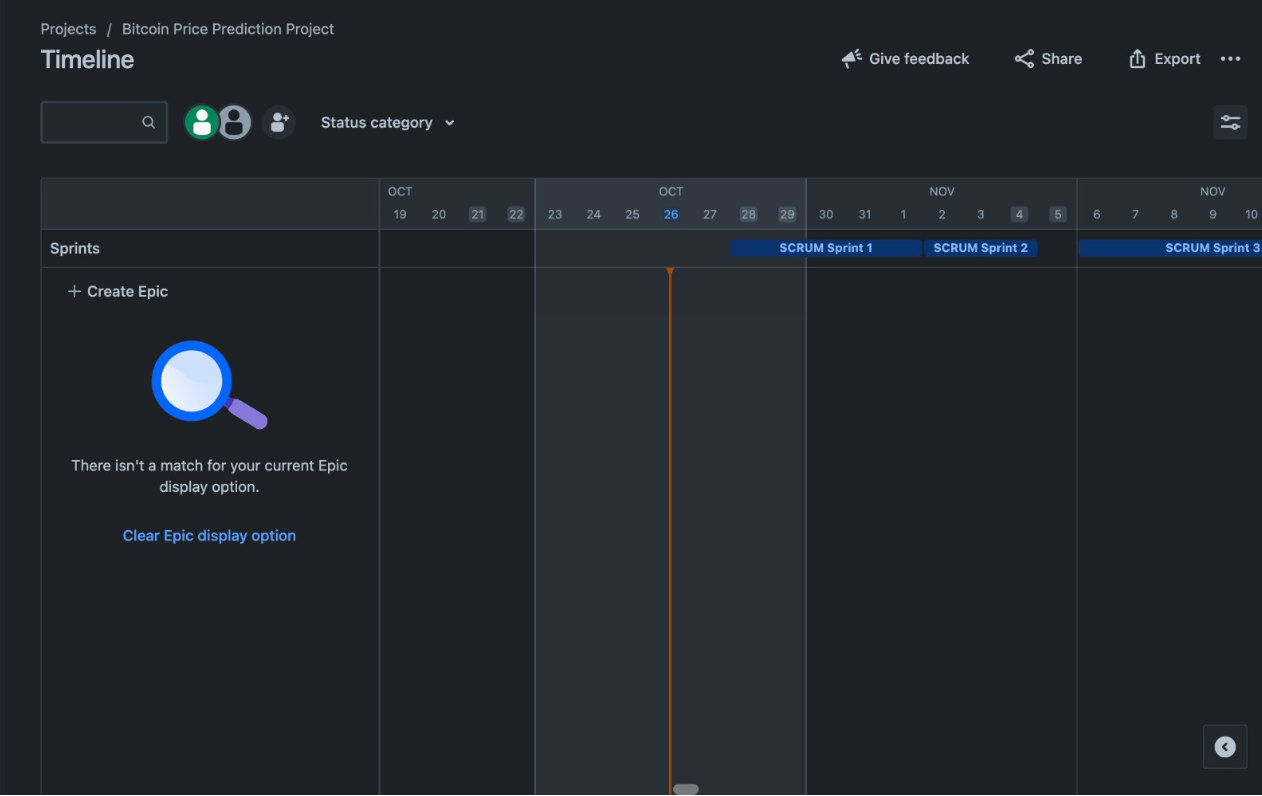
Solution Architecture:



1. Project Planning and Scheduling







1. **Coding and Solutioning**

🡪Phase one –Setting up the environment

* In this phase we created an separate anaconda environment for the project, in-order to avoid any clashes in requirements, then we installed the FbProphet, Yfinnance, PlotLy, Flask and other libraries using ‘pip install’ command.

🡪Phase two – Data collection

* Once the environment was ready, our first task was to collect the historical data

🡪Phase three – Data pre-processing and data visualization

* In this phase we visualized the time series data and cleaned the data and made it ready for the algorithm

🡪Phase four – Model building

* Once the data was ready it was time to feed the data to our FbProphet model in order to train it, this is what was done in this phase

🡪Phase five – Deployment

* After the model was ready we finally deployed the model using HTML,CSS and Flask

Let’s now look at these phases one by one in detail

**Setting up the environment**

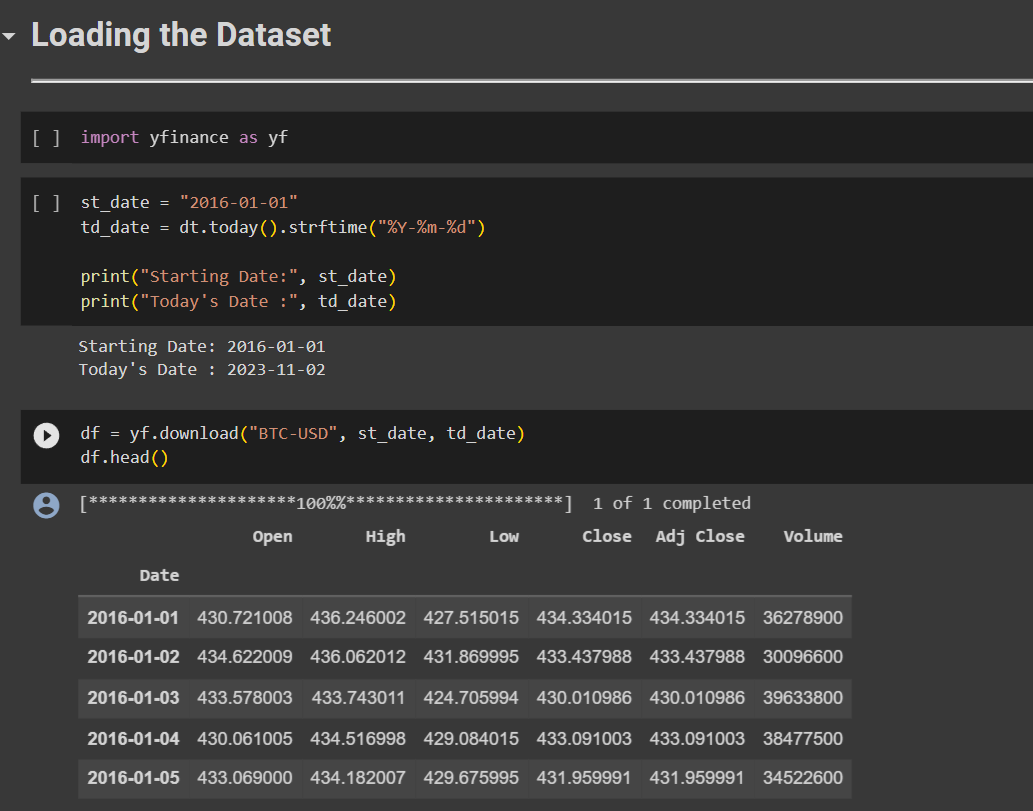
🡪This phase was simple and straight forward, we started by downloading anaconda navigator, then installed the Jupyter notebook.

🡪Once this was done we began installing required libraries starting from FbProphet, then PlotLy, Flask and Yfinance, we accomplished using pip commands.

**Data collection**

🡪The data collected in this project comes directly from [Yahoo Finance](https://finance.yahoo.com/quote/BTC-USD/history?p=BTC-USD) .

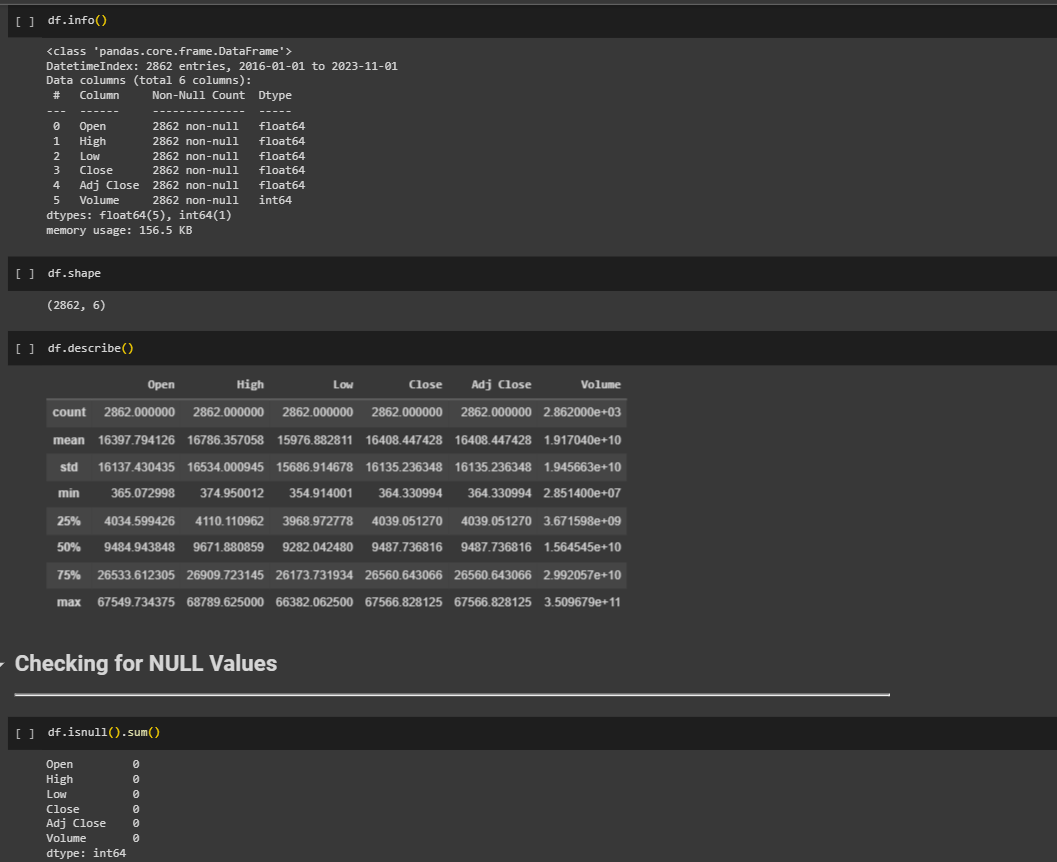
🡪To collect the data we have used Yfinance library of Python, the data is collected from 01/01/2016 till 02/11/23.



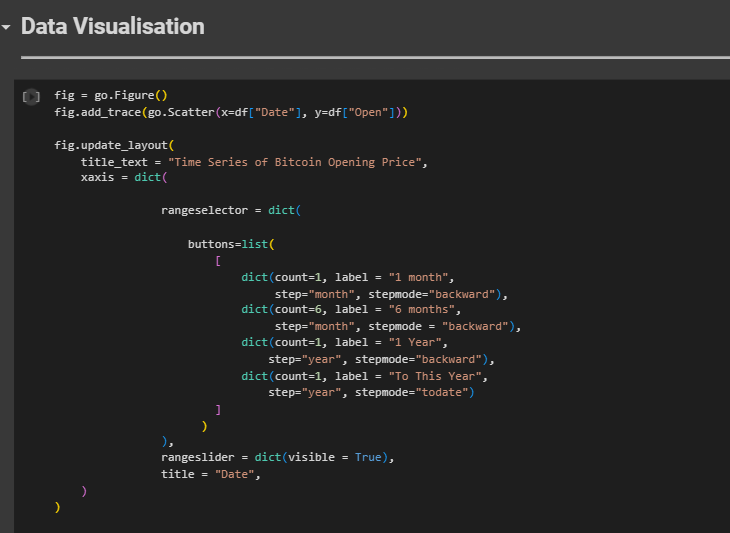
**Data pre-processing and visualization**

🡪This phase included understanding, visualizing and finally cleaning the data for model to use.

🡪First we checked weather there were any discrepancies.

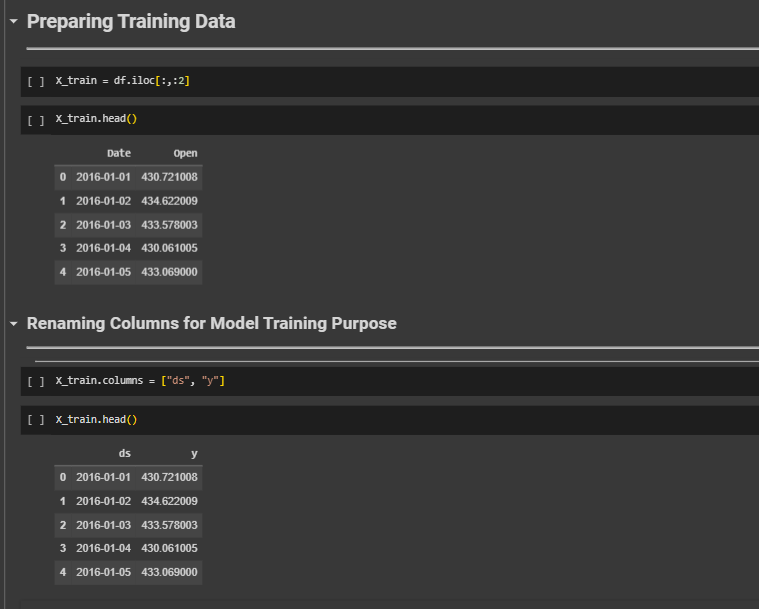


🡪After checking it was reviled that the there were some discrepancies in indexing of data which we fixed using ‘reset index’ function of ‘pandas’ apart from that there wasn’t anything that could hamper our accuracy of our model, so we started visualizing data to understand it better.





🡪In the final step of this phase we prepared the data for training and testing, we separated our target variable from rest of the dataset in-order for timeseries prediction to work.

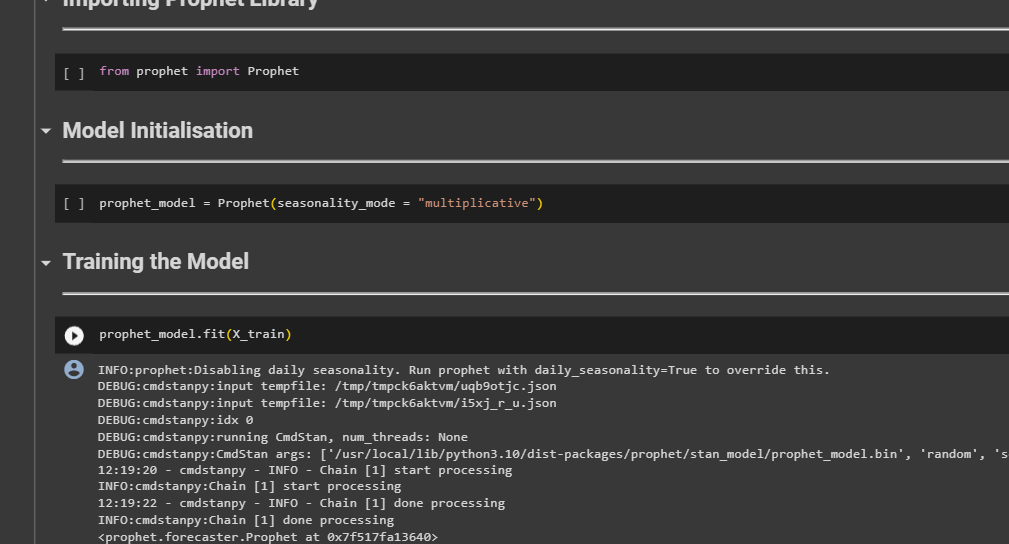


**Model building**

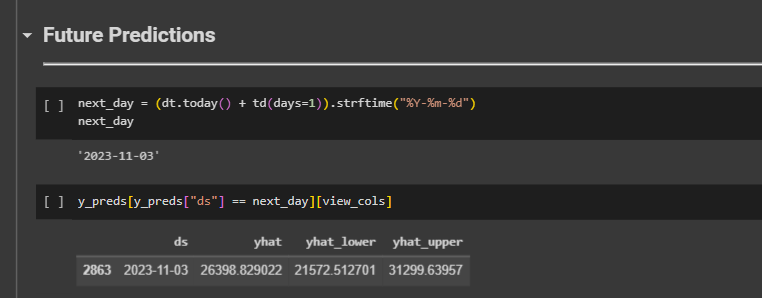
🡪Once we had understood the data, cleaned it and separated target variable, it was time for us to actually build the model

🡪So first we imported the model, then we created an ‘instance’ of that model.

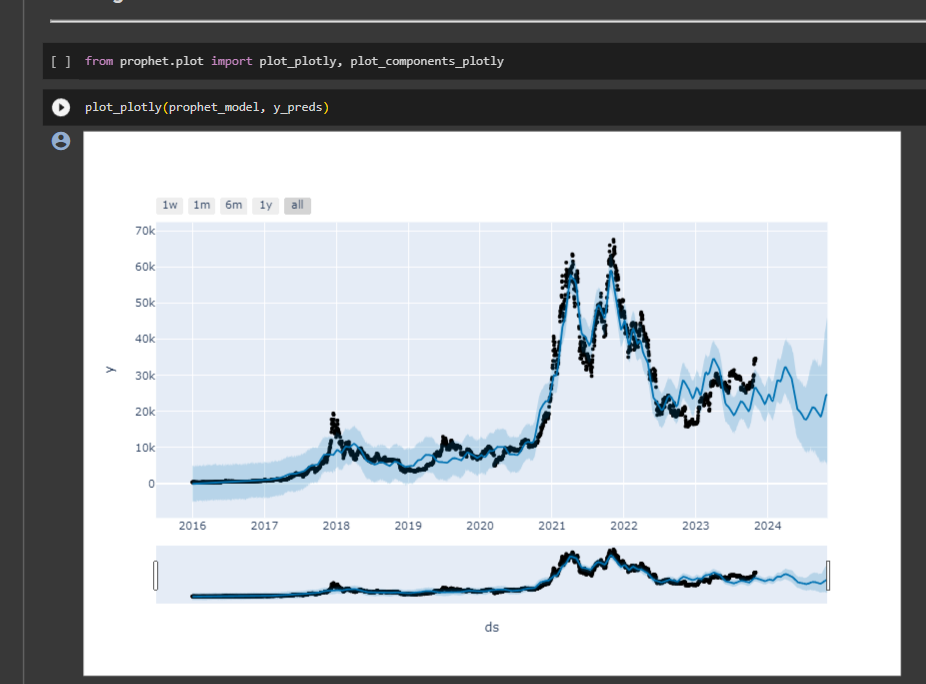
🡪After which we called ‘fit’ object from the model to fit our dataset.

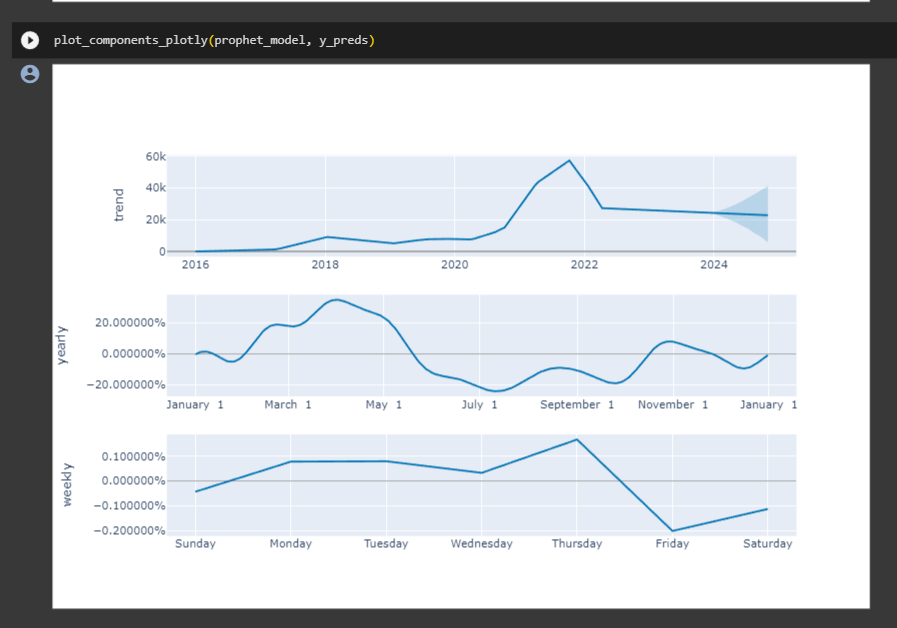


🡪Once the model was fitted it was time for us to test it and check if it gave reasonable predictions, we did that using ‘predict’ object of the model.

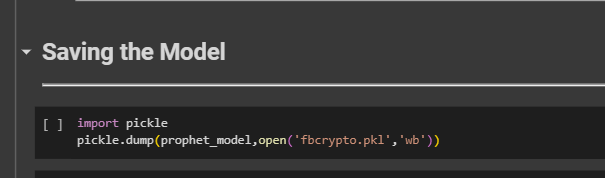


🡪After which we also plotted the future predictions of the model on the graph to understand it better.





🡪As the last step of this model we saved the it using ‘pickel’



**Model deployment**

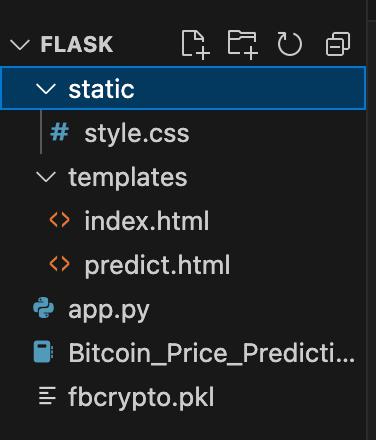
* + We are going to use Flask to deploy our machine learning model.
* Flask requires a saved version of our ipynb where our model is being developed and we are using pickle to save our model in

.pkl format.

àIn the flask application, the input parameters are taken from the HTML page These factors are then given to the model to predict the price of bitcoin on a selected date and showcased on the HTML page to notify the user. Whenever the user interacts with the UI and selects the “predict” button, the next page is opened where the user selects the date and predicts the output.

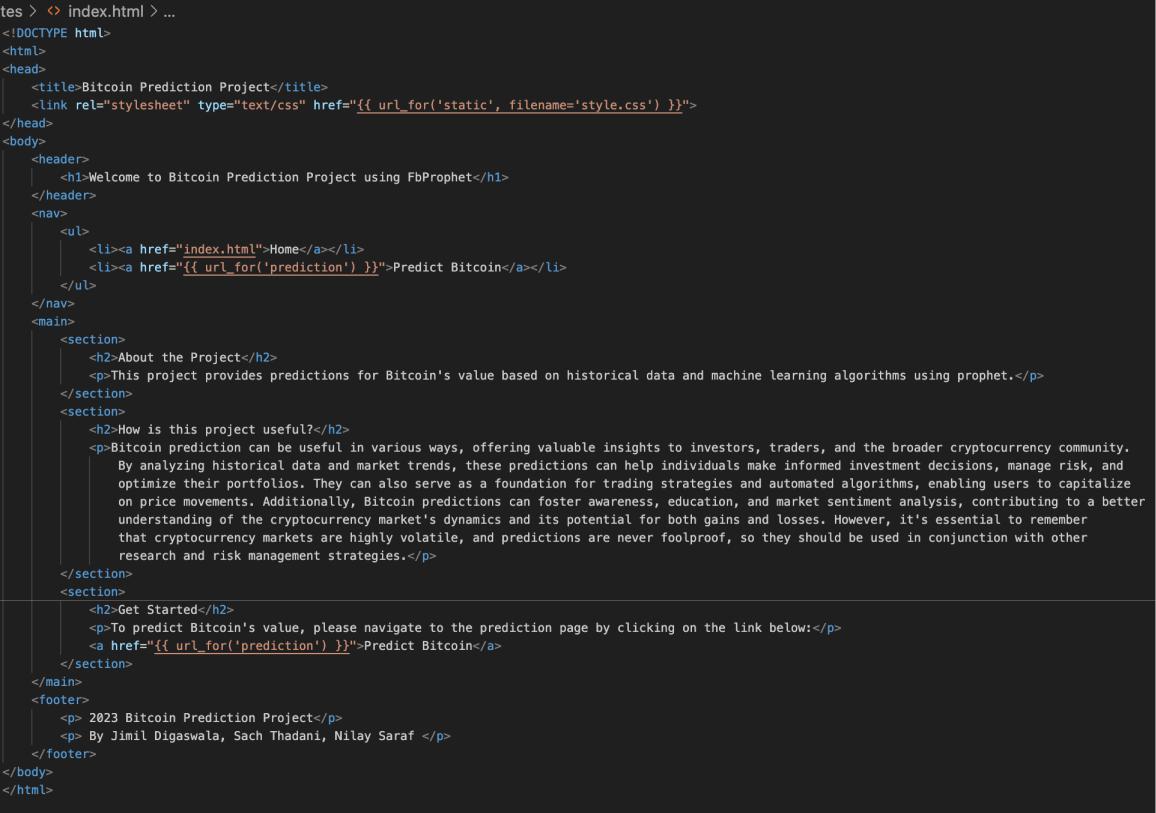
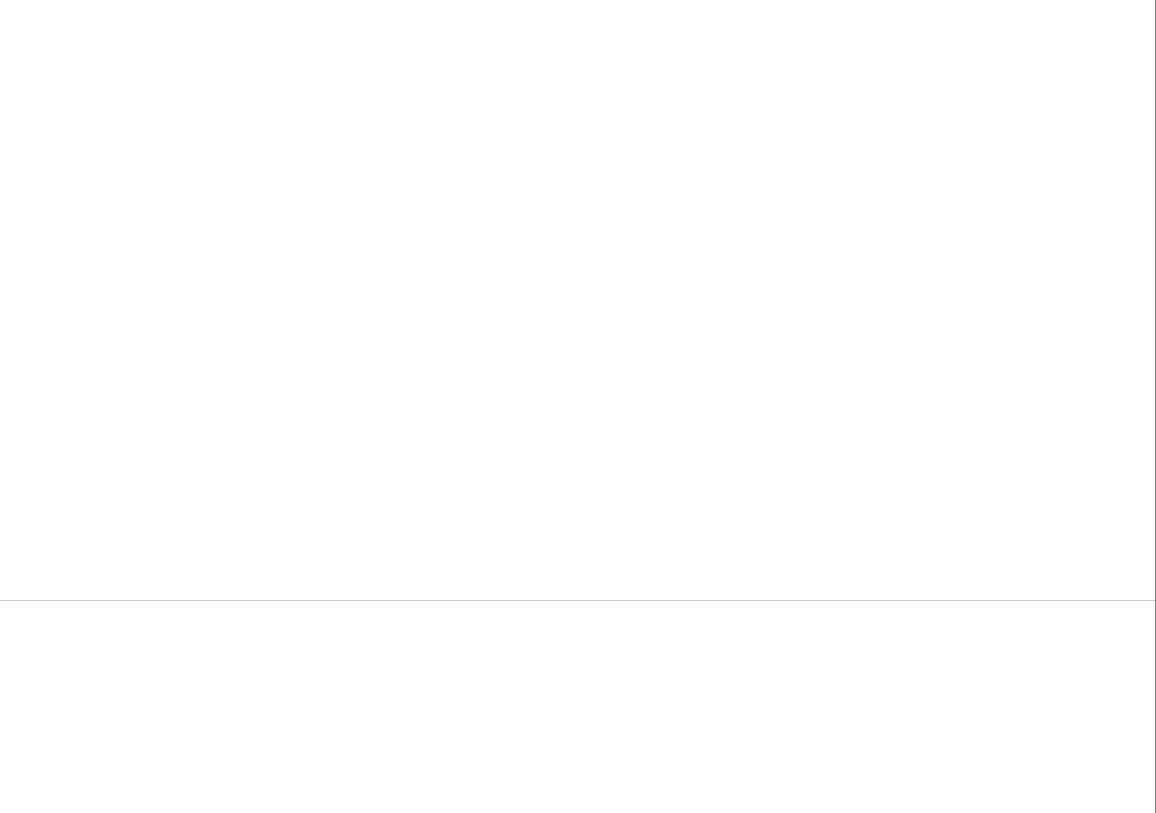


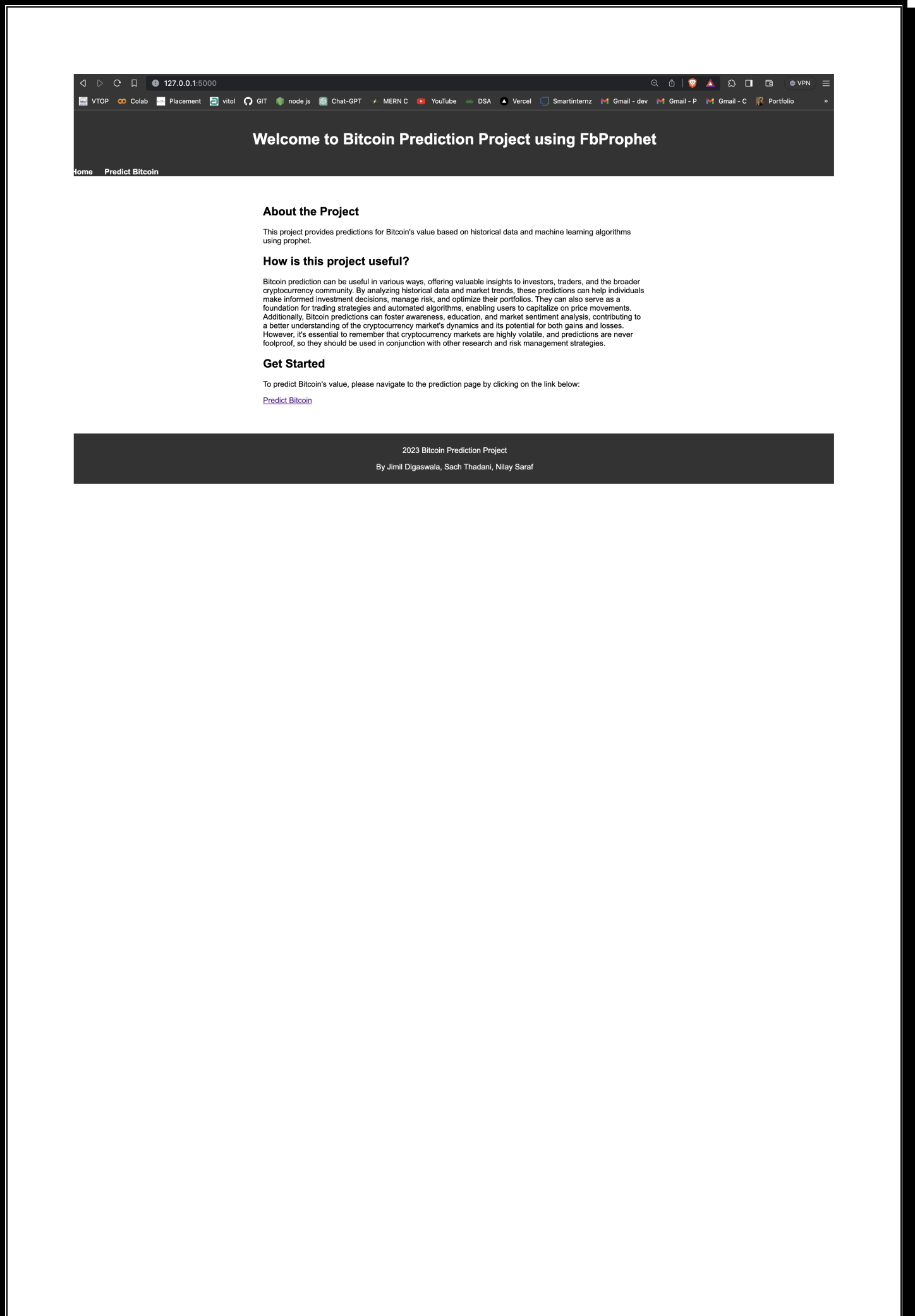
The file structure of our flask deployment looks like this:



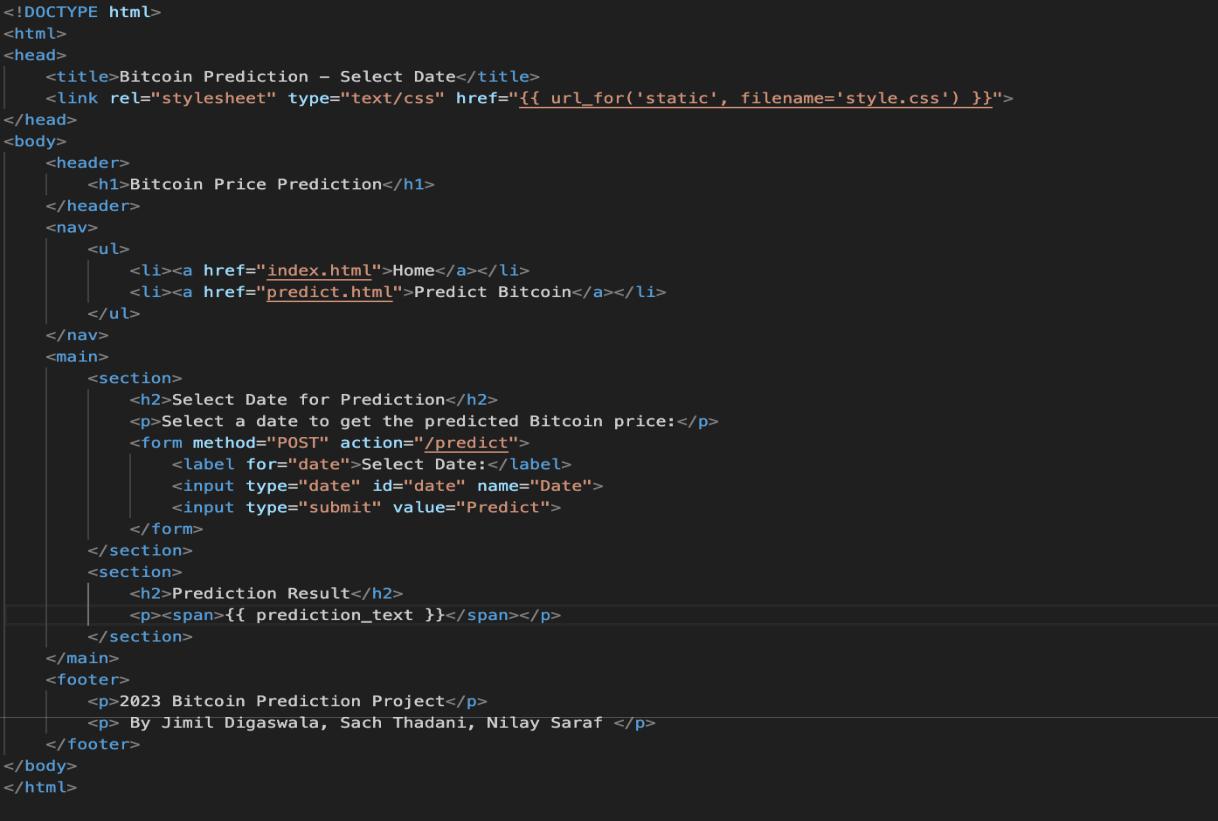
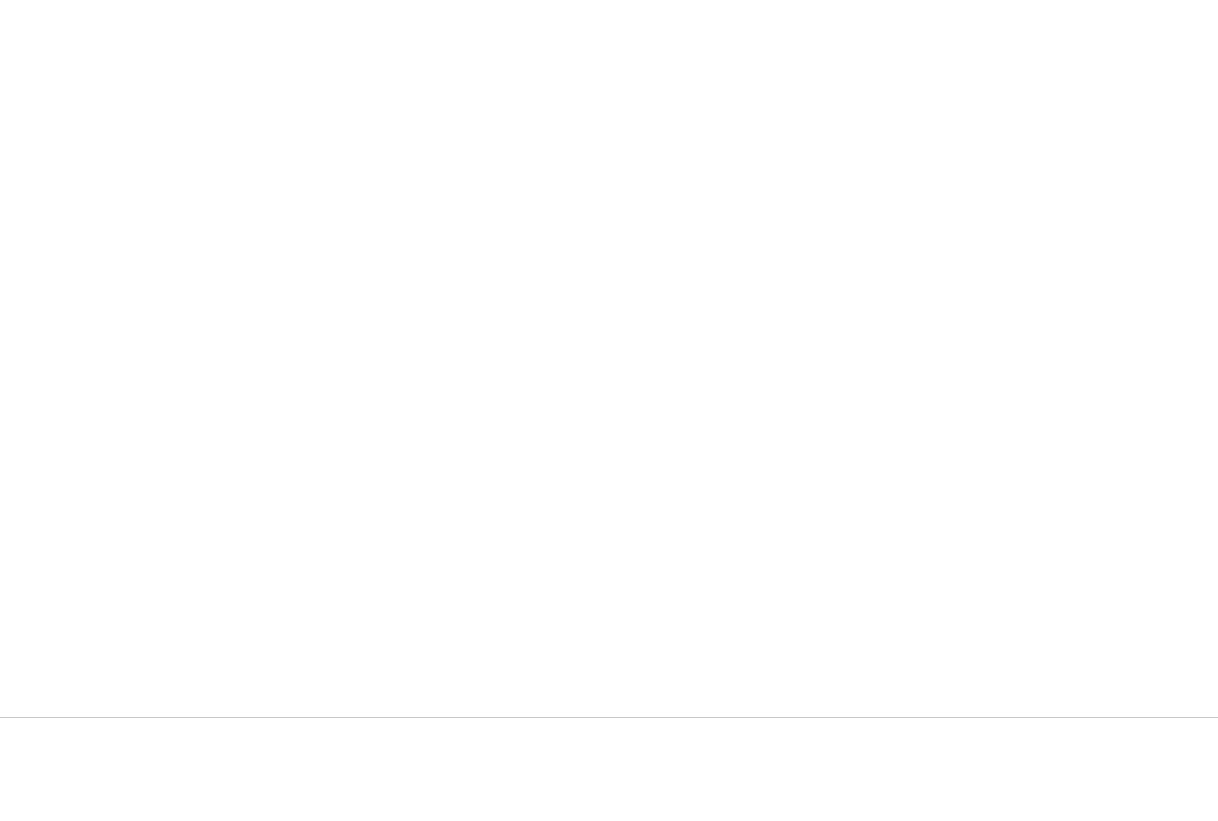
àWe have made two html files, index.html for the homepage and predict.html for the prediction page.

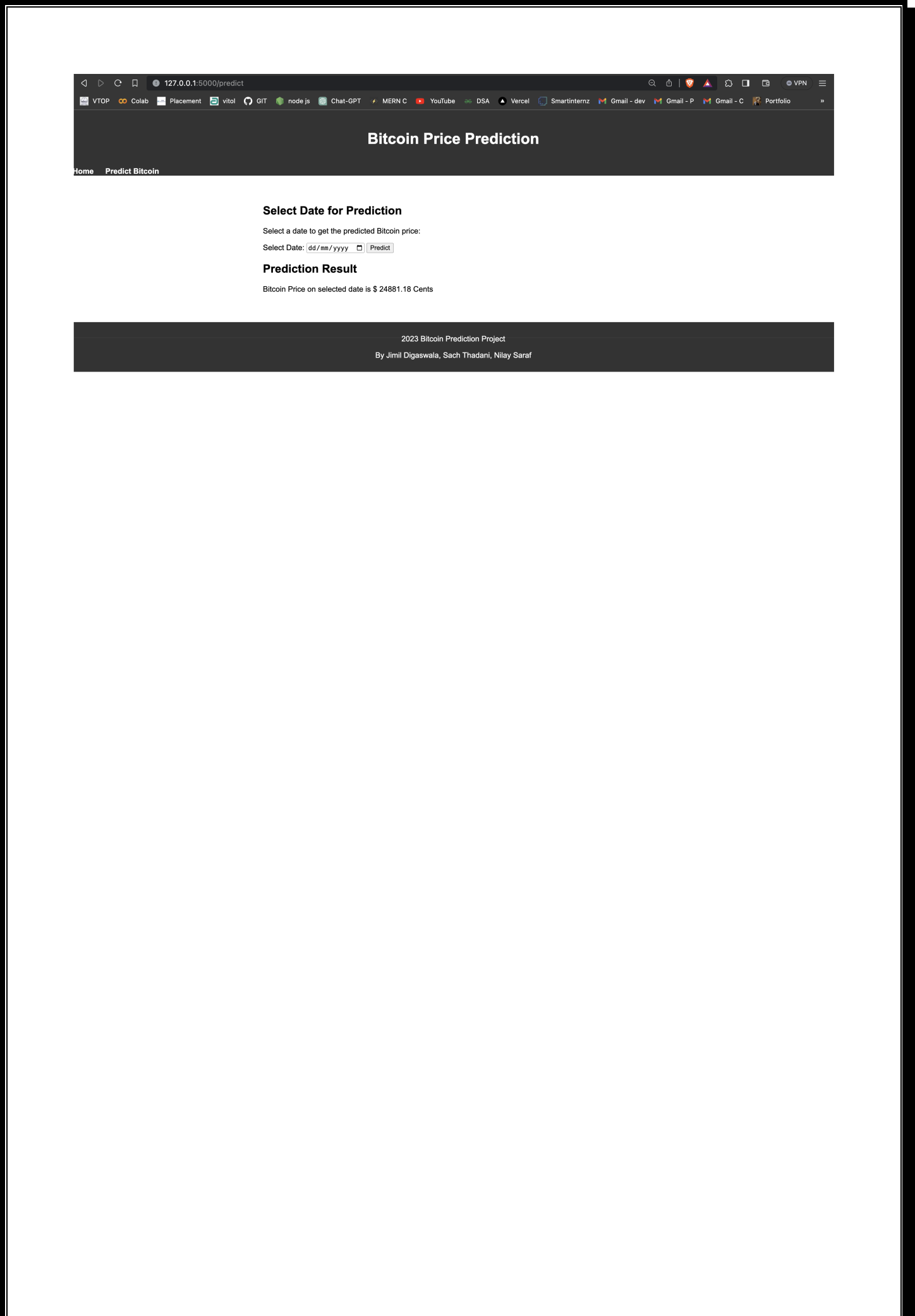
* The code and deployment of the **index.html** looks like this:





* The code and deployment of the **predict.html** looks like this:





Now, the main file of flask app.py is required to host the model as it contains the code which links the model, and the two html files responsible for ui and calling methods.

App.py code:

Task1: Importing libraries

import numpy as np



import pandas as pd



from flask import Flask, request, render\_template



import pickle

Task2: Creating our model and loading out model using pandas.

app = Flask(\_\_name\_\_)



m = pd.read\_pickle("fbcrypto.pkl")



Task3: Routing to HTML pages and making future predictions:

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



def index():



return render\_template('index.html')



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



def prediction():



if request.method == 'POST':



ds = request.form['Date']



ds = str(ds)



next\_day = ds



future = m.make\_future\_dataframe(periods=365)



forecast = m.predict(future)



prediction = forecast[forecast['ds'] == next\_day]['yhat'].item()



prediction = round(prediction, 2)



print(prediction)



return render\_template('predict.html', prediction\_text="Bitcoin Price on selected date is $ {} Cents".format(prediction))



return render\_template('predict.html')



if \_\_name\_\_ == "\_\_main\_\_":



app.run(debug=False)

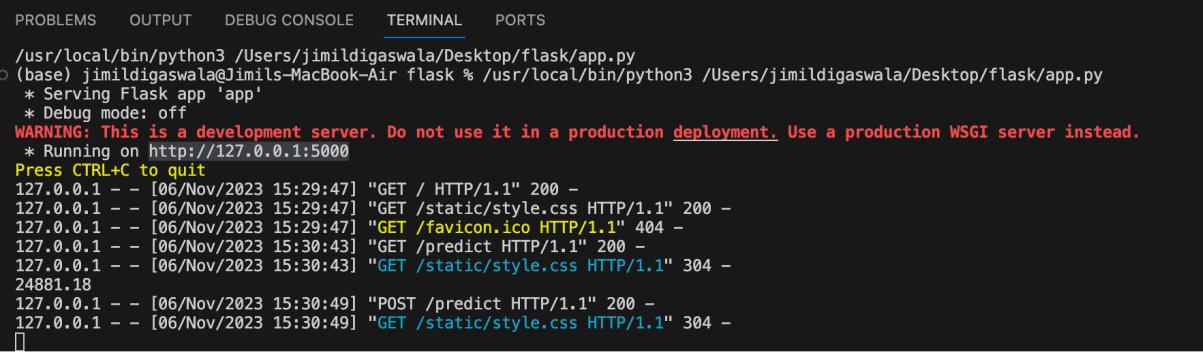


This is done using prophet model and the predictions made by the model is showed on the ui by routing in the html pages and then showing it in the deployment.

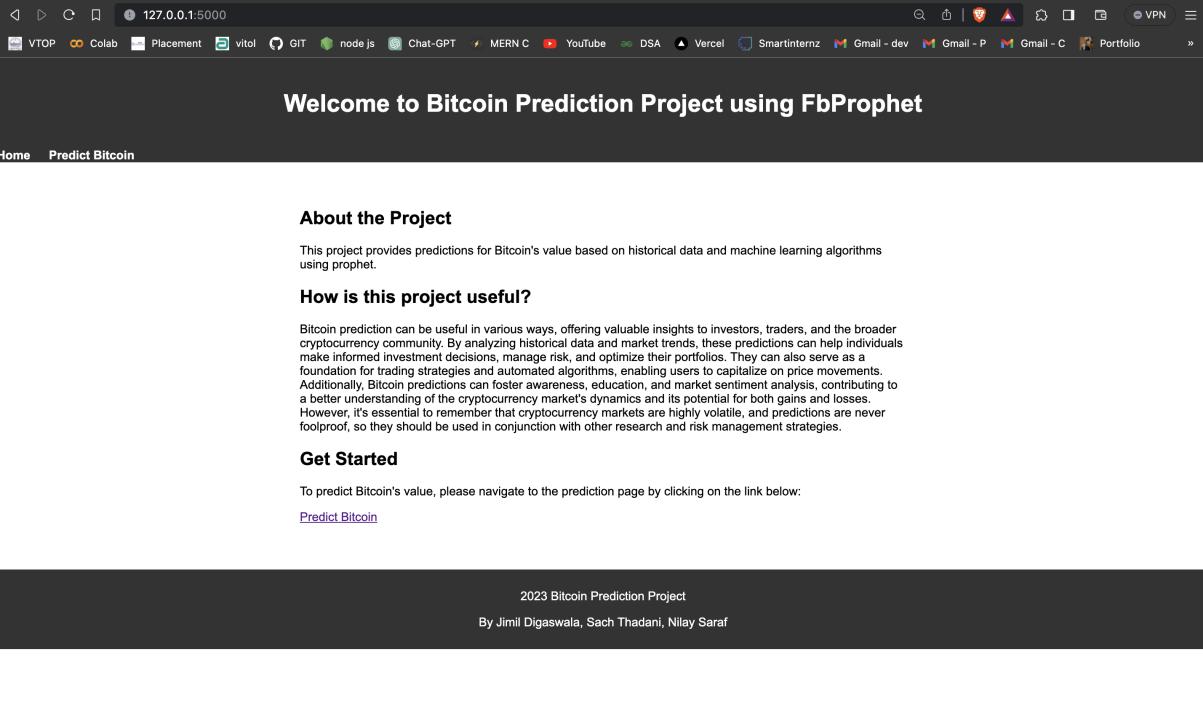
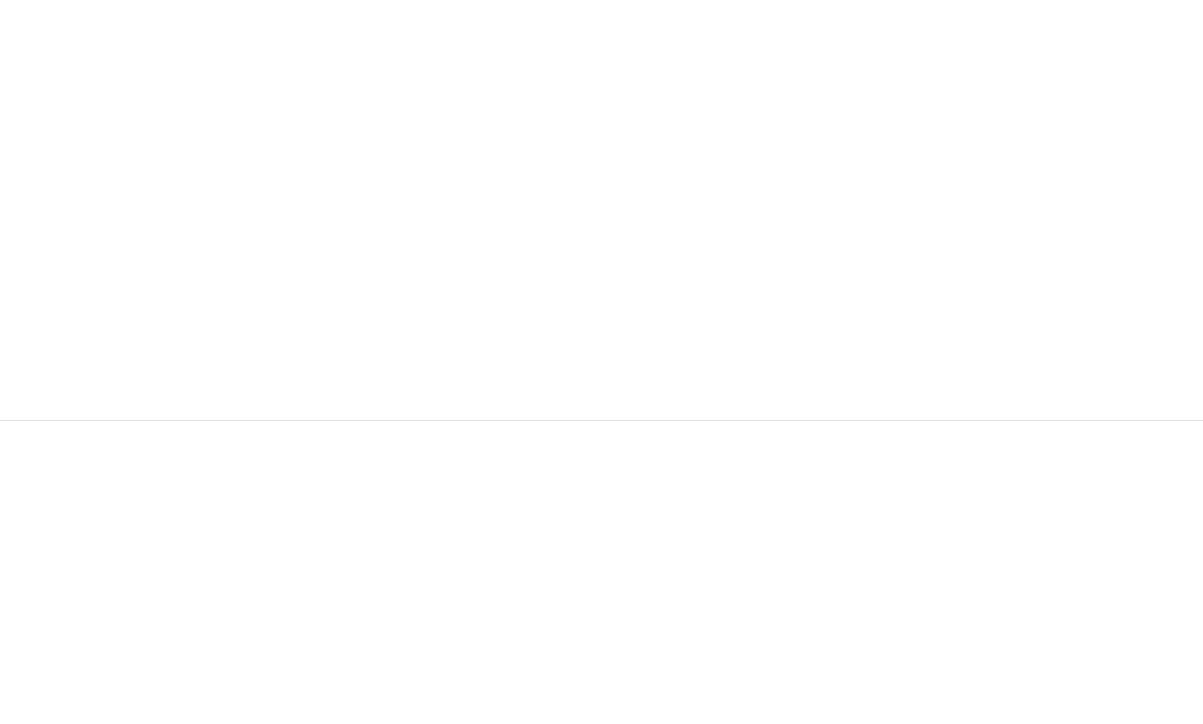


**Steps to predict the bitcoin price using the ui:**

1. Run the app.py file



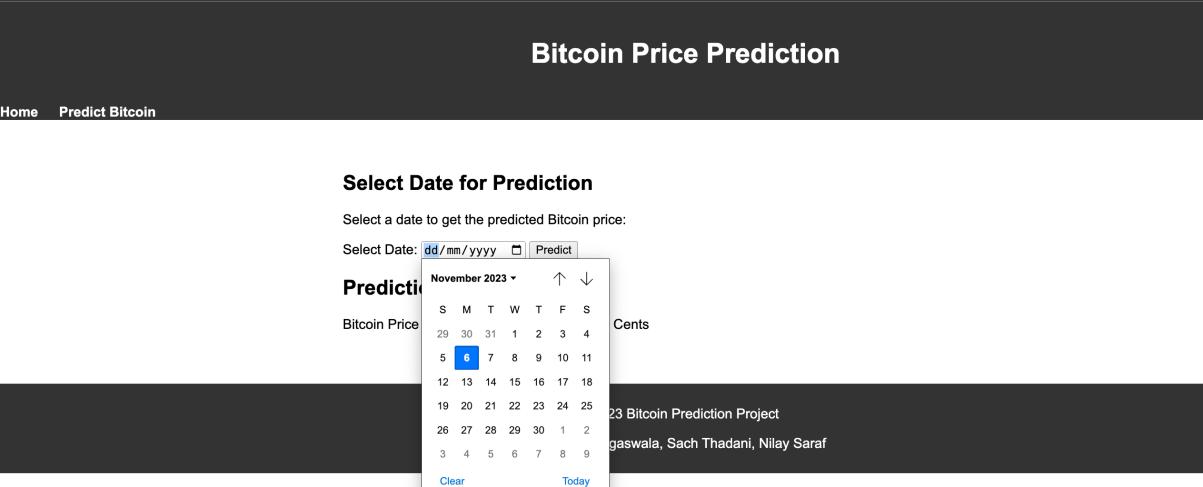
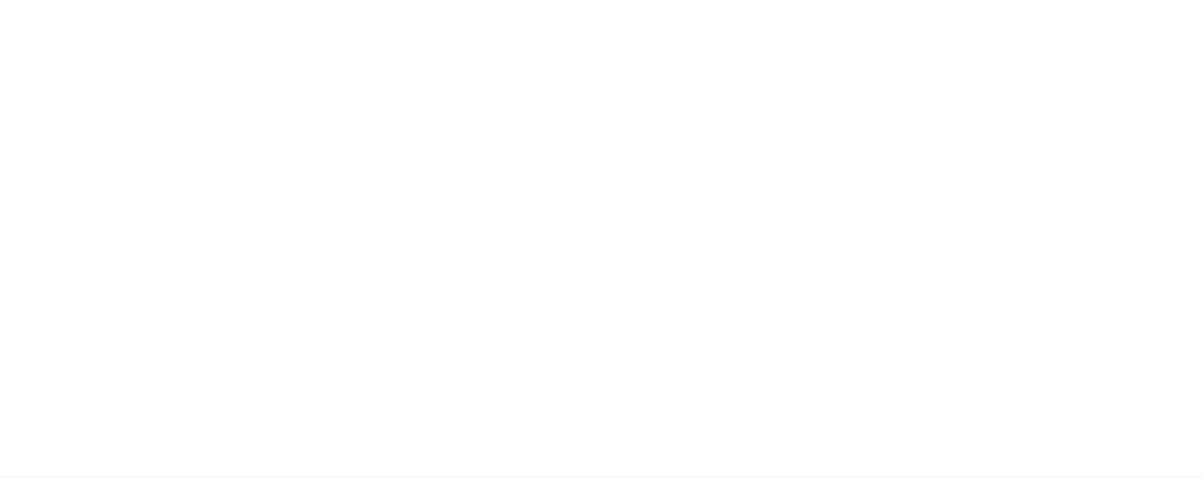
1. A localhost link will be provided, navigate to the link and you will see the page given below:



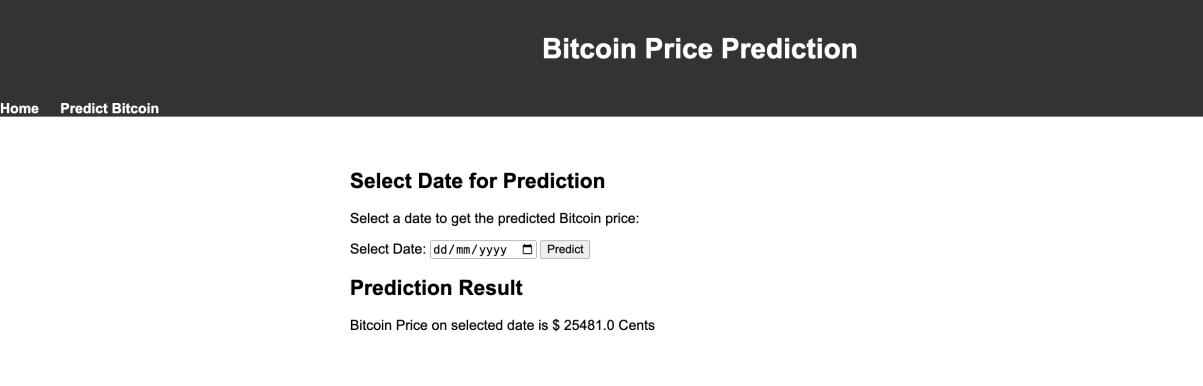
Now, click on the predict button and you will be navigated to the prediction page.



1. In the predict page, you will find a dialog box to select the date you want to predict the bitcoin price for:



1. After selecting the date, click on the predict button and the predicted price should be displayed on the screen.



1. **Performance Testing:**

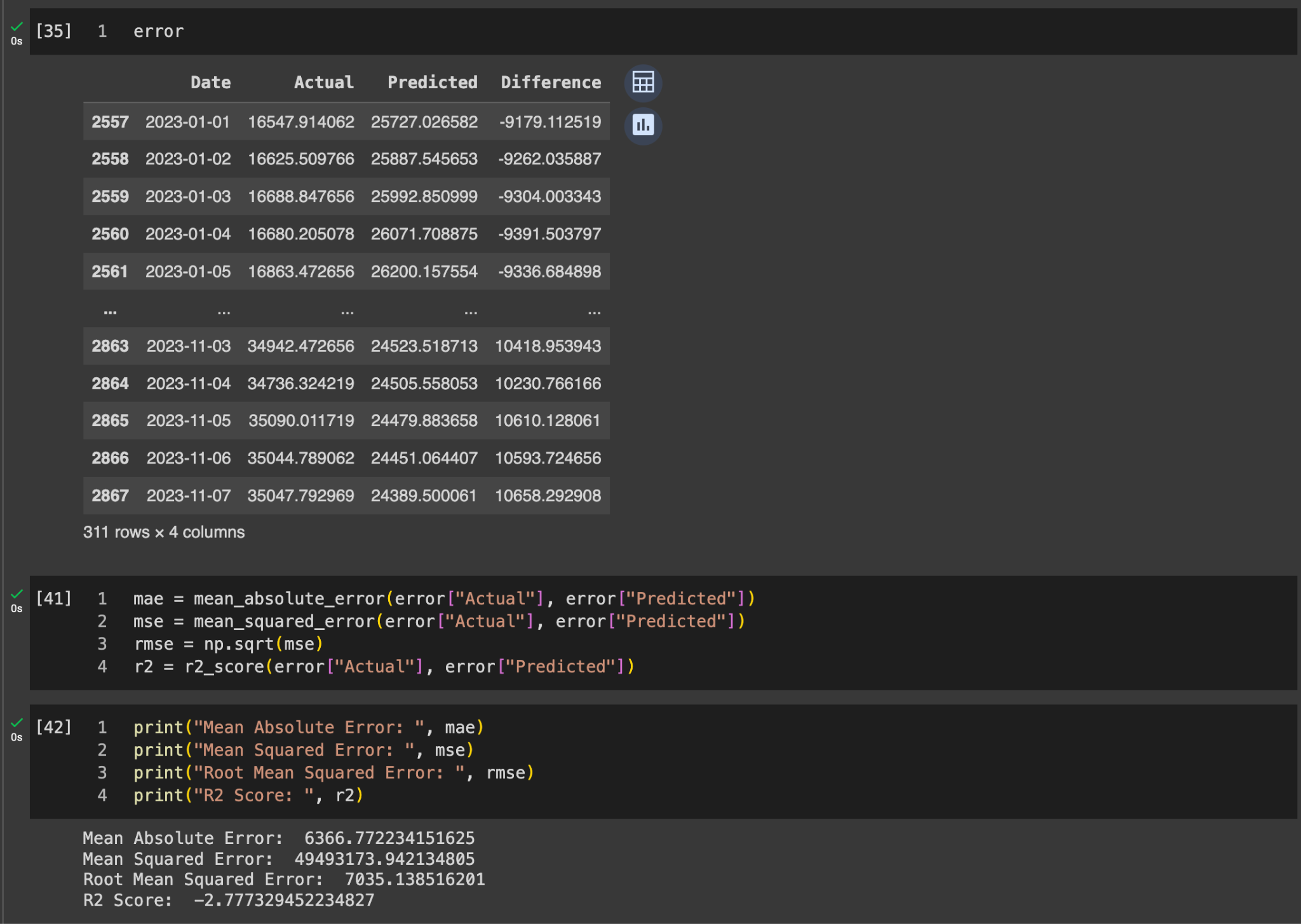
**Metrics:**

MAE: 6366.8

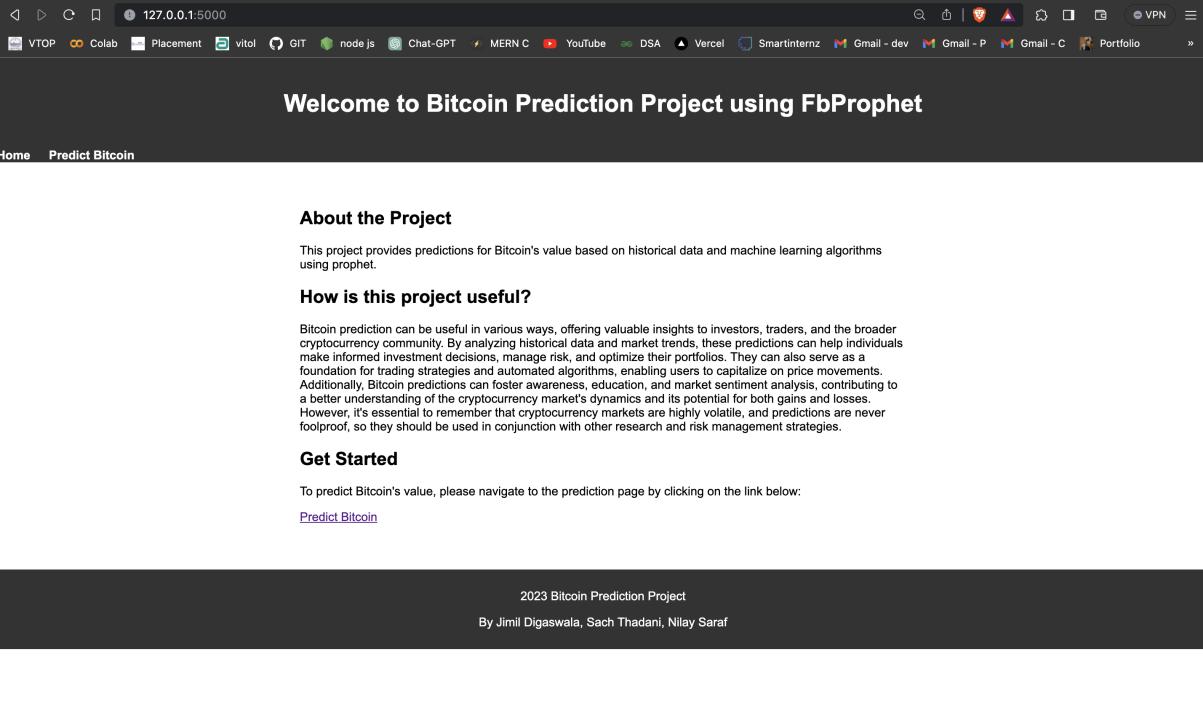
MSE: 49493173.95

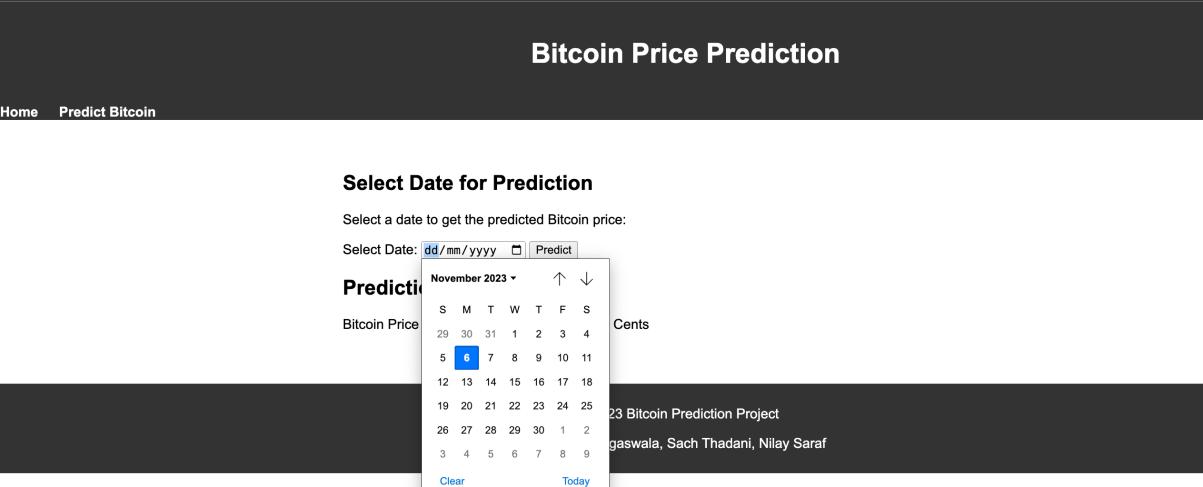
RMSE: 7835.14

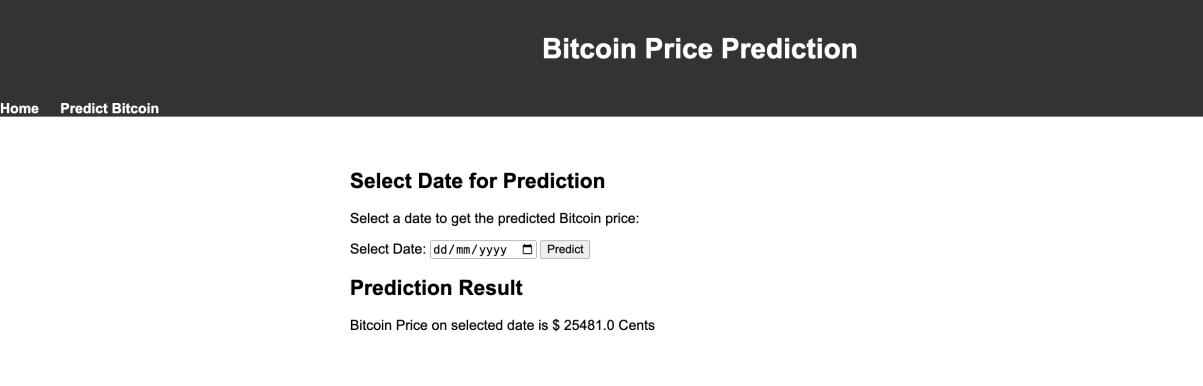
R2: -2.78



1. **Output Screenshots :**







1. **Advantages and Disadvantages:**

**Advantages:**

**Ease of Use:** FbProphet is designed to be user-friendly, making it easy for developers and analysts to implement time series forecasting without extensive expertise.

**Handling Missing Data**: FbProphet can handle missing data and outliers effectively, which is crucial when working with financial time series data that may have irregularities.

**Automatic Seasonality Detection:** The model can automatically detect and incorporate seasonality patterns in the data, simplifying the process for users who may not be familiar with the underlying seasonality components.

**Interpretability:** FbProphet provides insights into the forecast components, making it easier for users to interpret and understand the factors influencing the predicted prices.

**Flexibility in Input Data:** The model can handle both daily and seasonal data, providing flexibility in capturing different patterns and trends in the cryptocurrency market.

**Open Source and Community Support:** Being an open-source library, FbProphet benefits from community contributions and continuous improvement, ensuring updates and enhancements.

**Scalability:** FbProphet is scalable and can handle large datasets, making it suitable for applications with substantial historical financial data.

**Disadvantages:**

**Sensitivity to Hyperparameters:** The model's performance can be sensitive to the choice of hyperparameters, and tuning them for optimal results may require domain-specific knowledge.

**Limited Forecast Horizon:** FbProphet may not perform as well when forecasting for a very long-term horizon. Its strength lies in short to medium-term predictions.

**Assumption of Additive Components:** The model assumes that different components (trend, seasonality, holidays) contribute additively to the overall forecast. In cases where this assumption doesn't hold, the model's accuracy may be compromised.

**Lack of Advanced Features:** FbProphet may lack some advanced features available in other time series forecasting models, limiting its applicability in certain scenarios.

**Handling Rapid Market Changes:** Cryptocurrency markets can experience rapid changes and unexpected events. FbProphet may struggle to adapt quickly to sudden shifts in market dynamics.

**Limited Uncertainty Estimation:** While FbProphet provides uncertainty intervals, the estimation might be oversimplified, and more advanced models could offer more nuanced uncertainty estimates.

**Dependency on Historical Data Quality:** The accuracy of the predictions heavily relies on the quality and representativeness of the historical data. Inaccuracies or biases in the data can affect the model's performance.

1. **Conclusion:**

In conclusion, the crypto price prediction project employing FbProphet has successfully navigated through crucial phases, showcasing a robust framework for forecasting Bitcoin prices. The establishment of a dedicated Anaconda environment and the collection of comprehensive historical data from Yahoo Finance underscore the project's technical foundation. Through meticulous data pre-processing and visualization, the team ensured data integrity and visual clarity, laying the groundwork for effective model training. The implementation of FbProphet as the forecasting tool, combined with Flask for web deployment, reflects a holistic approach to time series prediction, demonstrating the team's technical proficiency. While achievements include user-friendly interfaces and insightful predictions, challenges such as hyperparameter sensitivity and limitations in forecasting horizons have been acknowledged. The project's learning outcomes span technical skills in data analysis, time series forecasting, and model deployment. Looking ahead, considerations for model refinement, exploration of advanced features, and continuous evaluation underscore the project's commitment to adaptability in the dynamic realm of cryptocurrency markets. In essence, this project not only contributes to the understanding of crypto market dynamics but also serves as a practical demonstration of machine learning applications in the financial domain, with implications for future advancements in predictive modelling.

1. **Future Scope:**

The future scope of this cryptocurrency price prediction project is promising and multifaceted. Firstly, continuous refinement and optimization of the forecasting model can enhance its accuracy and reliability, offering more precise predictions in an ever-evolving crypto market. Exploring advanced machine learning techniques, incorporating additional data sources, and leveraging ensemble models could further elevate the project's predictive capabilities. Additionally, integrating sentiment analysis from social media and news platforms may provide valuable insights into external factors influencing cryptocurrency prices. The project's scalability allows for the incorporation of multiple cryptocurrencies, expanding its utility across a broader spectrum of digital assets. Furthermore, collaboration with financial institutions, traders, and policymakers could yield valuable feedback and foster the project's integration into real-world financial decision-making processes. As the crypto space continues to grow and mature, the project holds the potential to contribute to a more informed and predictable market, promoting wider acceptance and understanding of digital assets among the general public.

1. **Appendix:**

**GitHub link:**

https://github.com/smartinternz02/SI-GuidedProject-589580-1697519158.git

**Demonstration video link:**

https://drive.google.com/file/d/1P0kfZ8UDmD4X3-pgpn52Eh3StG0sH8Rh/view?usp=sharing