Mini Project report on

# GOLD RATES PREDICTION

**By**

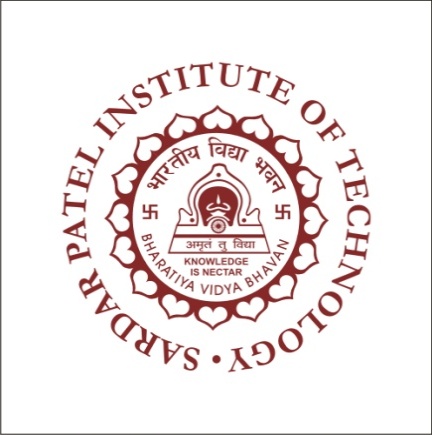
**OTHNIEL ALEXANDER**

**PRADNYOT MORWEKAR**

Under the guidance of

**Internal Supervisor**

**Prof. PALLAVI THAKUR**

****

Department of Master Of Computer Applications

Sardar Patel Institute of Technology

University of Mumbai

2018-19

**CERTIFICATE OF APPROVAL**

This is to certify that the following students

**OTHNIEL ALEXANDER**

**PRADNYOT MORWEKAR**

Have satisfactorily carried out work on the project entitled

“**Gold Rates Prediction**”

Towards the fulfillment of Mini Project, as laid down by University of Mumbai during year 2018-19.

Project Guide

(Prof. Pallavi Thakur)

**PROJECT APPROVAL CERTIFICATE**

This is to certify that the following students

**OTHNIEL ALEXANDER**

**PRADNYOT MORWEKAR**

Have successfully completed the Mini Project report on **“Gold Rates Prediction*”***, which is found to be satisfactory and is approved

At

SARDAR PATEL INSTITUTE OF TECHNOLOGY,

ANDHERI (W), MUMBAI.

INTERNAL EXAMINER EXTERNAL EXAMINER

Head of Department Principal

|  |  |
| --- | --- |
| **INDEX** | |
| **CONTENTS** | **PAGE NO.** |
| **Abstract** | **5** |
| **Objective** | **5** |
| **List of Tables** | **6** |
| **List of Figures** | **6** |
| **1. Introduction** | **7** |
| **1.1** Problem Definition | **7** |
| **1.2** Objective & Scope | **7** |
| **1.3** Existing System | **8** |
| **1.4** Proposed System | **8** |
| **1.5** System Requirements | **9** |
| **2. Literature Survey** | **10** |
| **3. Software Requirement Specification[SRS] &Design** | **11** |
| **3.1** Introduction | **11** |
| **3.2** Overall Description | **11** |
| **3.2.1** Product Functions | **13** |
| **3.2.2** User Characteristics | **13** |
| **3.3** Specific Requirement | **14** |
| **3.3.1** Functional Requirements | **14** |
| **3.3.2** Non-Functional Requirements | **14** |
| **4. Project Analysis & Design** | **15** |
| **4.1** Methodologies Adapted | **15** |
| **4.1.1** Detailed Life Cycle of Project | **16** |
| **4.2** Modules | **16** |
| **4.2.1** UML Diagram-Use Case Diagram with Report | **17** |
| **4.3** Database Schema/Design | **18** |
| **4.3.1** E-R Diagram | **18** |
| **4.3.2** DFD Diagram | **19** |
| **5. Project Implementation & Testing** | **20** |
| **5.1** Gantt Chart | **20** |
| **5.2** Snapshots of UI | **23** |
| **5.3** Test Cases | **31** |
| **6. Limitations** | **33** |
| **7. Future Enhancements** | **33** |
| **8. Refrences** | **34** |
|  |  |
|  |  |
|  |  |

**Abstract**

Historically, gold had been used as a form of currency in various parts of the world including USA. In present times, precious metals like gold are held with central banks of all countries to guarantee re-payment of foreign debts, and also to control inflation which results in reflecting the financial strength of the country.

Recently, emerging world economies, such as China, Russia, and India have been big buyers of gold, whereas USA, South Africa, and Australia are among the big seller of gold.Forecasting rise and fall in the daily gold rates, can help investors to decide when to buy (or sell) the commodity. But Gold prices are dependent on many factors such as prices of other precious metals, prices of crude oil, stock exchange performance, Bonds prices, currency exchange rates etc.

We in this project would forecast gold rates using the most comprehensive set of features and would apply various machine learning algorithms for forecasting and compare their results. We also identify the attributes that highly influence the gold rates.

**Objectives**

Following are the objectives of the Web Application.

* It helps investors to decide when to buy or sell a commodity.
* To enable commoners to easily invest in gold.
* To inform investors or any users about the gold prices including all the details.
* It can predict prices of gold up-to next four years.
* To inform people about the changes in gold prices.
* To predict prices of gold in a particular month.
* To enable discussions among the investors about the drop and hike in the gold prices with the help of a story telling graph.

|  |  |  |
| --- | --- | --- |
| **List of Figures** | | |
| **Figure No.** | **Figure Name** | **Page No.** |
| 2.1 | Block Diagram | 15 |
| 4.1 | RAD Model | 16 |
| 4.2 | Use-Case Diagram | 17 |
| 4.3.1 | E-R Diagram | 17 |
| 4.3.2 | Data Flow Diagram | 19 |
| 5.2 | Gantt Chart | 20 |
| 5.3 | WBS Chart | 18 |
| 5.4 | Pert Chart | 19 |

|  |  |  |
| --- | --- | --- |
| **List of Tables** | | |
| **Table No.** | **Table Name** | **Page No.** |
| 1.5.1 | Hardware Requirements | 8 |
| 1.5.2 | Software Requirements | 19 |
| 4.2 | Use-Case Tables | 19 |
| 5.3.1 | Test Case Tables | 31 |

**1. Introduction**

**1.1. Problem Definition**

The challenge of this project is to accurately predict the future adjusted closing price of Gold ETF across a given period of time in the future. The problem is a regression problem, because the output value which is the adjusted closing price in this project is continuous value.

**1.2. Objective And Scope**

## **Objectives of the system:**

## The proposed system will be online system, so it will not be time-consuming.

### Investors would easily be able to view the the articles created on the blog site,add their own articles which enables thorough discussion.

### Security is maintained. No one will able to see the unauthorized data without registering, also the admin can add or delete stories which ensures that no fake articles are not introduced.

### Story telling prediction graphs will be generated quickly.

## **Scope of the proposed system:**

### The system has been developed to be simple and user-friendly. Therefore, even a non-technical user(investor) would be able to use the system effectively and without any issues.

### Since it is upgrade-able, further enhancements are possible.

### The system is able to maintain speed and accuracy.

**1.3. Existing System**

* Many few investors own platform(Mobile Application or Web Application) that acts as a complete package providing features from prediction to discussion.
* Others are operating through Whatsapp messages(communication and discussion) and financial apps providing information about the prices of the current day. Keeping track of all these and predicting the prices on our own is tedious task.
* These Web Applications are less user friendly and do not present the results in form of graphs telling the user stories.

**1.4. Proposed System**

* Web Application that lets users make quick account(SignUp).
* Visitor can view articles but can’t create or discuss without an account (Login).
* Users will get the prediction of the gold prices for the desired month.
* User can either view articles or create their own articles and graphs.
* Users can discuss and review the articles with other investors but only by login.
* Users can view average of Gold ETF prices (1 ETF = 1 gram of gold)
* Users can view variety of articles created by other investors.
* Administrator can create or delete graphs and discussion articles.
* Only the owner of the article can edit and delete its contents(articles/graphs).

**1.5. System Requirements:**

**Hardware Requirements:**

|  |  |
| --- | --- |
| **Processor** | Dual Core Processor or Above |
| **RAM** | 512MB RAM |
| **Storage** | 10GB Hard Disk Space for smooth run |

**Software Requirements:**

## **Back-End:**

### Django with mpld3

## **Front-End:**

* JetBrains Pycharm and Jupyter notebook

## **Others:**

### JavaScript Frameworks like EJS, Passport, Route Handlers.

### Bootstrap v3.3.7

### Cascading Style Sheets (CSS)

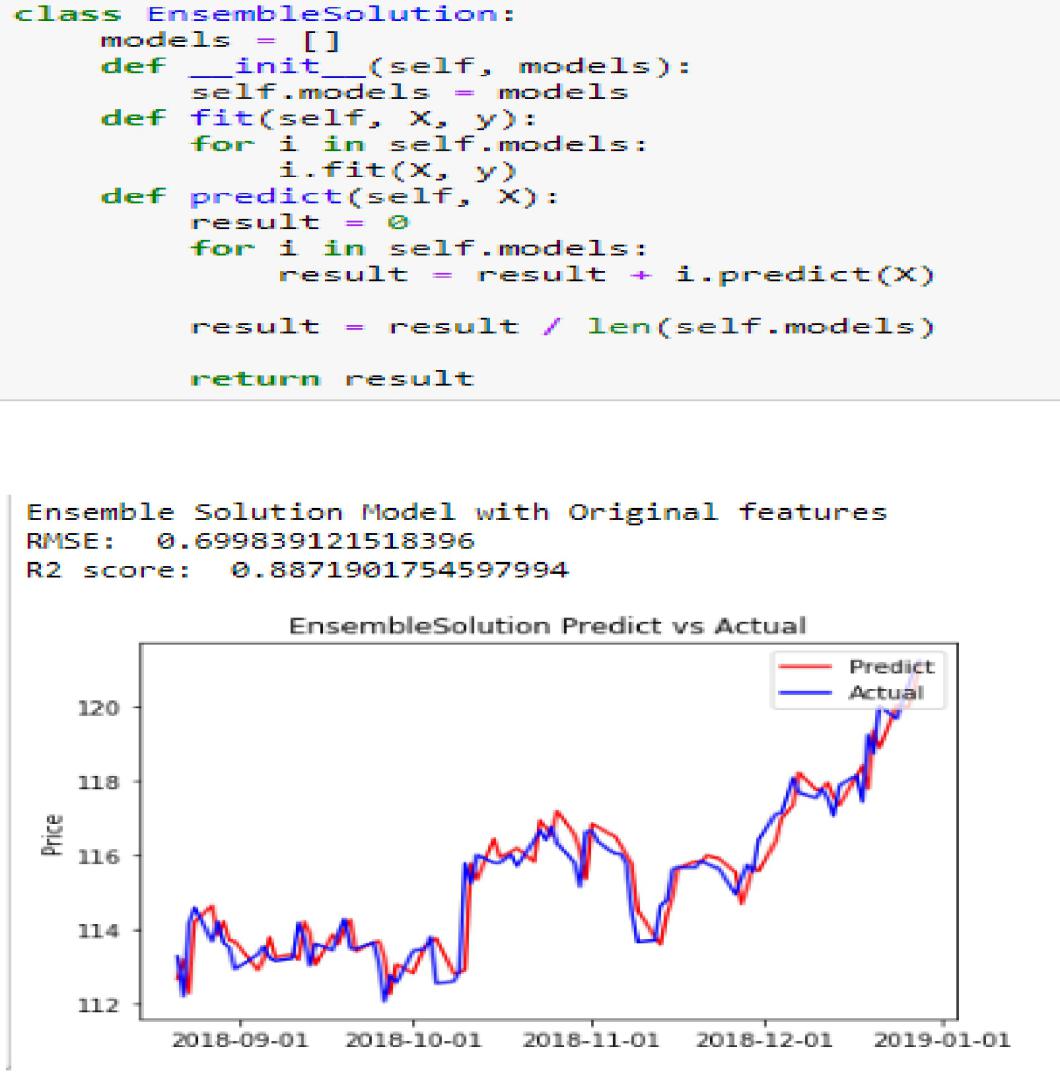
### Font Awesome

### Hyper Text Mark-up Language (HTML)

**2. Literature Survey**

**Linear Regression Models:**

* Instead of using one particular model as done in the past projects, here Linear Regression model is used.
* Ensemble model merges the top three performing models(three in our case).
* Linear Regression solution with all features shows best results (with RMSE 0.699 and R2 score 0.887) in comparison with other solution models.



JetBrains PyCharm IDE for writing, running and debugging code:

* PyCharm is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment" \o "Integrated development environment) (IDE) used in [computer programming](https://en.wikipedia.org/wiki/Computer_programming" \o "Computer programming), specifically for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)" \o "Python (programming language)) language. It is developed by the Czech company [JetBrains](https://en.wikipedia.org/wiki/JetBrains" \o "JetBrains). It provides code analysis, a graphical debugger, an integrated unit tester, integration with [version control systems](https://en.wikipedia.org/wiki/Revision_control" \o "Revision control) (VCSes), and supports web development with [Django](https://en.wikipedia.org/wiki/Django_(web_framework)" \o "Django (web framework)).
* Coding assistance and [analysis](https://en.wikipedia.org/wiki/Code_analysis" \o "Code analysis), with [code completion](https://en.wikipedia.org/wiki/Autocomplete" \o "Autocomplete), syntax and error highlighting, [linter integration](https://en.wikipedia.org/wiki/Lint_(software)" \o "Lint (software)), and quick fixes
* Project and code navigation: specialized project views, file structure views and quick jumping between files, classes, methods and usages
* Python [refactoring](https://en.wikipedia.org/wiki/Refactoring" \o "Refactoring): including rename, extract method, introduce variable, introduce constant, pull up, push down and others.

**3. Software Requirement Specification (SRS) & Design:**

**3.1 Introduction:**

**Product Perspective:**

The software described in this SRS is the software for a complete gold rate prediction. The system merges various hardware and software elements and further interfaces with external systems. Thus, while the software covers the majority of the system's functionality, it relies on a number of external interfaces for persistence and unhandled tasks, as well as physically interfacing with humans.

Since this is a machine learning project we need to train the model using datasets which will enable the model to perform various actions.

**3.2 Overall Description:**

**3.2.1 Product Functions :**

The product functions include:

1. **Authentication:**

Users/investors are required to Sign-up and Log-in. Users can view articles but cannot create or discuss without signing in. System redirects users to login page if he/she tries to review,create,edit,discuss the article without log-in into the system.

1. **Viewing Articles(Graphs):**

System facilitates users viewing of various articles created by other users/investors with or without authentication.

1. **Creating Articles:**
2. Investors can create articles(graphs) and provide full details of it-name, month,year ,commodity , etc.
3. **Reviewing Articles:**

Investors/users can provide feedback and can review the articles created by themselves or other investors. This facility demands log-in into the system, means authenticate himself/herself.

5. Users/investors will get get the gold rate for the desired month.Also they will be provided with the average ETF (1 ETF = gram of gold) .

1. **Edit/Delete articles:**

Only the owner can edit/delete its content(articles, review, graphs). For security purpose apart from owner, admin can delete users review, articles and graphs.

**Software Interfaces:**

The interfaces setup an infrastructure by Incorporating required libraries Sklearn, Numpy, Pandas, Matplotlib and Seaborn.

The interfaces then perform dataset preparation which incorporates data from different sources, processes the data into pandas dataframe, the data is normalized by using the min\_max function.

The web application communicates with the server and plots the graph based on the user inputs.It does this, as the model is trained to perform such actions by using the past data.

**User interfaces:** If the user is not a first-time user, he/she should be able to see the articles page directly when the application is opened. Here the user chooses the type of search he/she wants to conduct. Every user should have a profile page where they can edit their e-mail address, phone number and password. Also, the user can set the Web application to his/her preferred language.

**3.2.2 User characteristics**:-

* Admin: Admin has the full access to the system which means he is able to manage any activity with regard to the system. He is the highest privileged user who can access to the system.

Key functions of admin are:

* Allocate resources
* Manage administrator charges
* Manage articles
* Manage users

**User**:

Interacts with the systems most often to supply service to customers.

* Keep track of campground details
* Keep track of availability of particular campgrounds.
* Keep track of billing details

**Constraints**:

* System is wirelessly networked with an encryption.
* Only administrator can access the whole system.
* Each user should have initial id and password.
* Database is password protected.

**Assumptions and dependency:**

* Each user must have a valid user id and password.
* Server must be running for the system to function.
* Users must log in to the system to access any article.
* Only the administrator can delete the articlesalong with the owner(user who created the article).

3.3 Specific Requirements:

3.3.1 Functional Requirements:

• The system will allow user to view Articles and its full details with or without Registration.

• The system will restrict the functionality of creating Articles, discussing and reviewing it to only registered users.

• It will provide the functionality where users can see the prediction of gold rates for the exact month.

• Users can edit/delete their own articles.

3.3.2 Non-functional Requirements:

1. **Usability**: The system provides a good and easy to use GUI for users to interact with the system.

2. **Security**: The users are required to authenticate themselves before impacting system. Impacting system means creating any articles, discuss, review.

It is the facility provided by system to prevent unauthorized access.

3. **Performance**: The system must have high performance rate when executing user’s input and should be able to provide response within short period of time. System is persistent. Data is directly stored onto the database and gives quick flash message to user.

4. **Availability**: The system must be available for access 24 hours, 7 days a week. Also in occurrence of any major malfunctioning, the system should online again as soon as possible. Also system will give the same performance even if many users log-on at the same time.

5. **Error Handling**: Errors must be considerably minimized and appropriate error message that guides the user to recover from error should be provided.

1. **Project Analysis And Design**

4.1 Detailed Life Cycle of the Project

Rapid Application Development (RAD) Model is used for designing this system.Unlike Waterfall methods, RAD emphasizes working software and user feedback over strict planning and requirements recording. In this model, we go through prototyping and collecting feedback iteratively until satisfactory product is developed. RAD requires system to following:

**Phase I:** Figure out the requirements

**Phase II:** Build Prototypes

**Phase III:** Get user feedback

**Phase IV:** Repeat Phase II and III (Sprints)

**Phase V:** Testing



Fig 4.1 Diagrammatic Representation of RAD Model

The process undertaken this project are as follows :

 1.Set Up Infrastructure

 a) I Python Notebook

 b)Incorporate required libraries Sklearn, Numpy, Pandas, Matplotlib and Seaborn

 c)Git Project Organization

 2.Dataset Preparation

 a)Incorporate data from different sources

 b)Process the data into pandas dataframe

 c)Normalize the data using Sklearn’s MinMax Function

 d)Timeseries Split with n\_splits=10

 3. Develop Benchmark Model

 a) Set up basic decision tree regressor with default parameters as benchmark model.



4. Develop Solution Model & improve using GridsearchCV

5. Linear Regression top three performing models

6. Evaluate all the solution models and benchmark model & Plot the result

7. Refine the models using Feature selection

8. Develop Benchmark, solution and ensemble models

9. Evaluate and compare the results with Original feature models

10. Plot, analyze and describe the results for report.

Gold Rates Prediction system has gone through following sprints:

10.1 Setting up infrastructure

10.2 Dataset preparation

10.4 Testing Front-end

10.4 Evaluate solution model and plot results

10.5 Testing the security and quality for robustness.

4.2 Use Case Diagram with Report

Use Case Description

**Actors:**

1. User

2. Organizer

3. Administrator

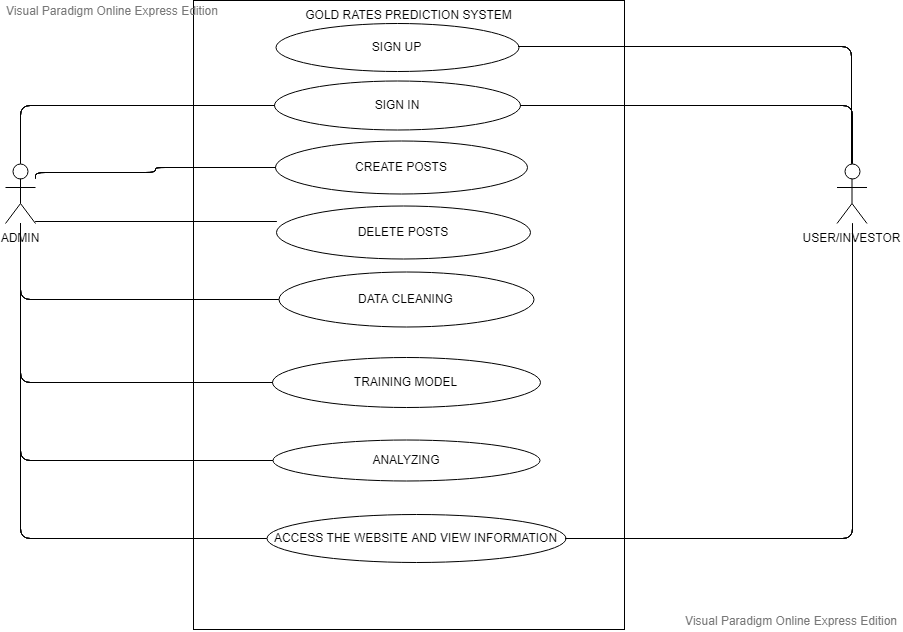


Fig 4.2 Use Case Diagram

**Use Cases:**

1. Sign-in

2. Create Posts

3. Delete Posts

4. Data Cleaning

5. Data Visualization

6. Training Model

7. Analyzing

8. Access the Website and View Information

# Table 4.2: Use Case Tables

|  |  |
| --- | --- |
| Use Case Id | 1 |
| Use Case Name | Sign-In |
| Actor | User, Admin |
| Pre-Condition | They must register themselves first. |
| Post-Condition | Users can view and review articles, Admin can control the activity. |
| Flow of events | Register, login, access system |

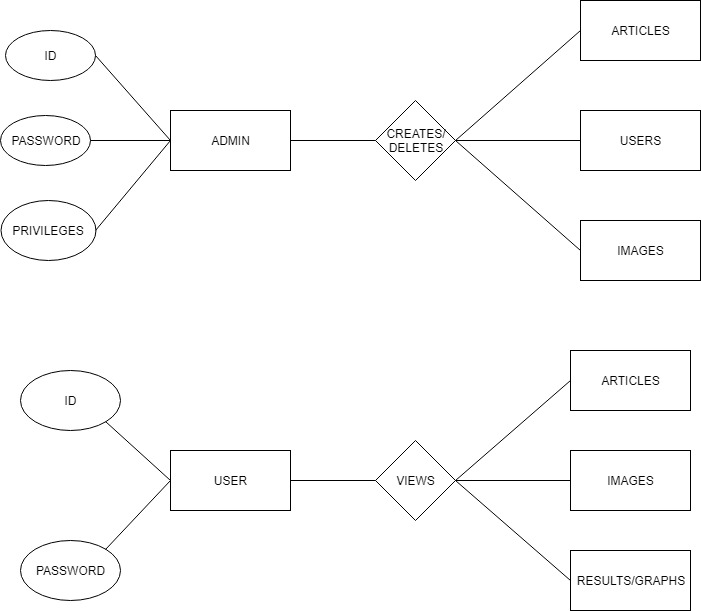
|  |  |
| --- | --- |
| Use Case Id | 2 |
| Use Case Name | Create Posts |
| Actor | Admin |
| Pre-Condition | Sign-in is necessary. |
| Post-Condition | Admin can can create, review and can delete. |
| Flow of events | Login, Create Articles |
| Use Case Id | 3 |
| Use Case Name | Delete Posts |
| Actor | Admin |
| Pre-Condition | Login is must. |
| Post-Condition | Admin can delete their reviews.  Other users, can view it. |
| Flow of events | Login, make review, and delete it. |

|  |  |
| --- | --- |
| Use Case Id | 4 |
| Use Case Name | Analyzing |
| Actor | Analyst |
| Pre-Condition | User must login |
| Post-Condition | Model Analyzed |
| Flow of events | Data cleaning,Data Visualization,Training Model,Analyzing |

|  |  |
| --- | --- |
| Use Case Id | 5 |
| Use Case Name | Access website and view information |
| Actor | User |
| Pre-Condition | Login is must. |
| Post-Condition | User as well as admin can view the information but only admin can delete it |
| Flow of events | Login, create, view, delete. |

4.3 Database Schema/Design

*4.3.1. E-R Diagram*

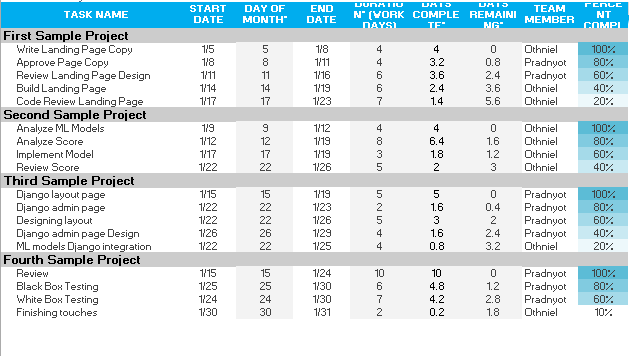


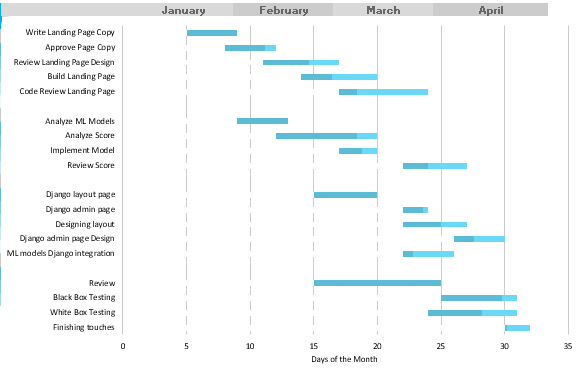
# *4.3.2 DFD Diagrams*

# DFD

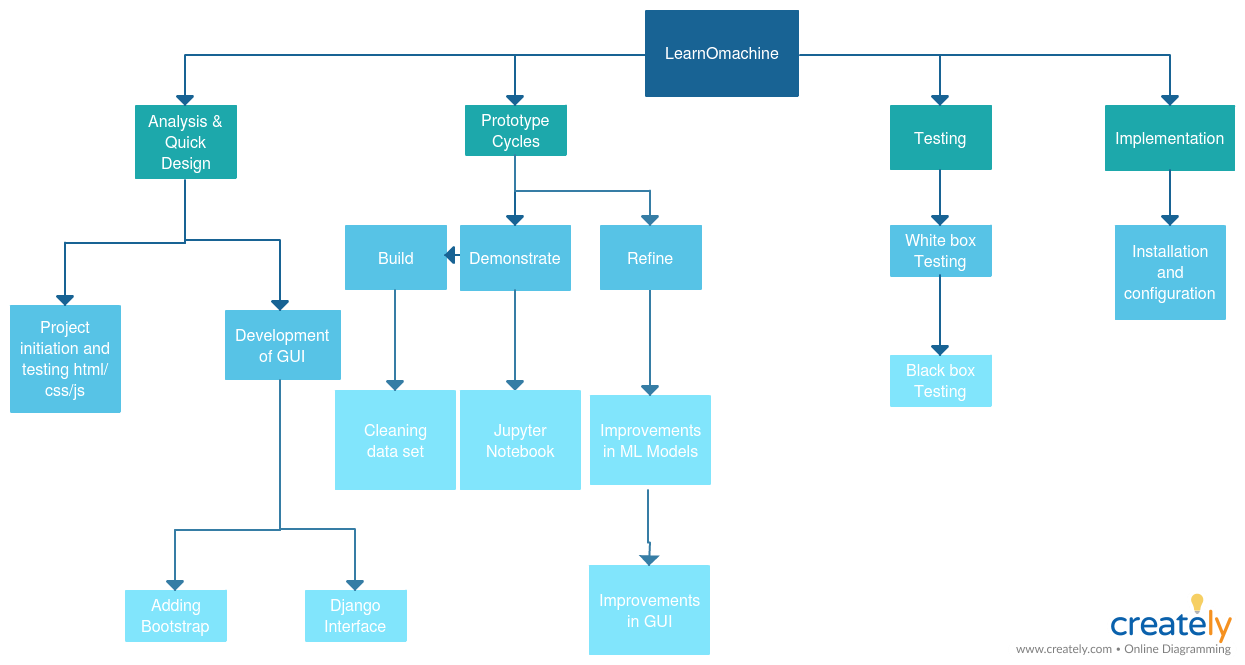
**5. Project Implementation and Testing**

5.1 Gantt Chart

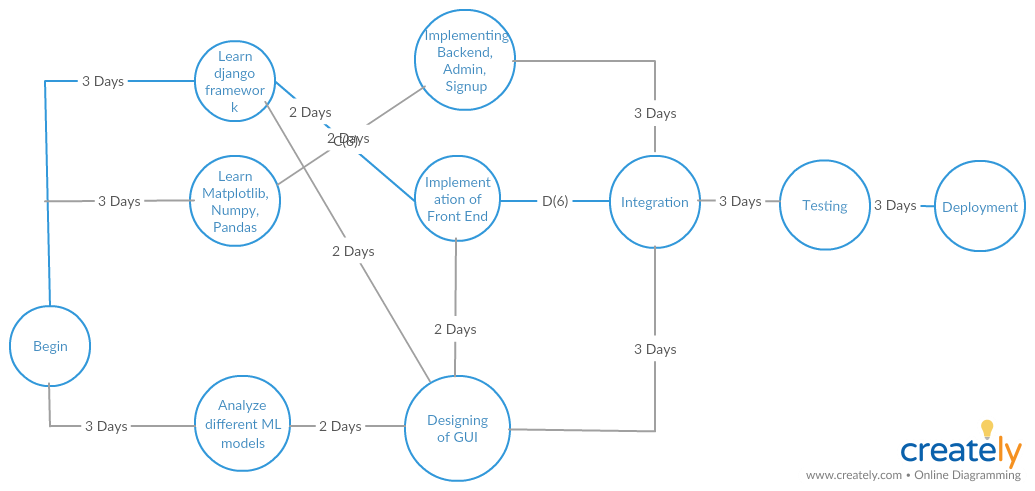
****

****

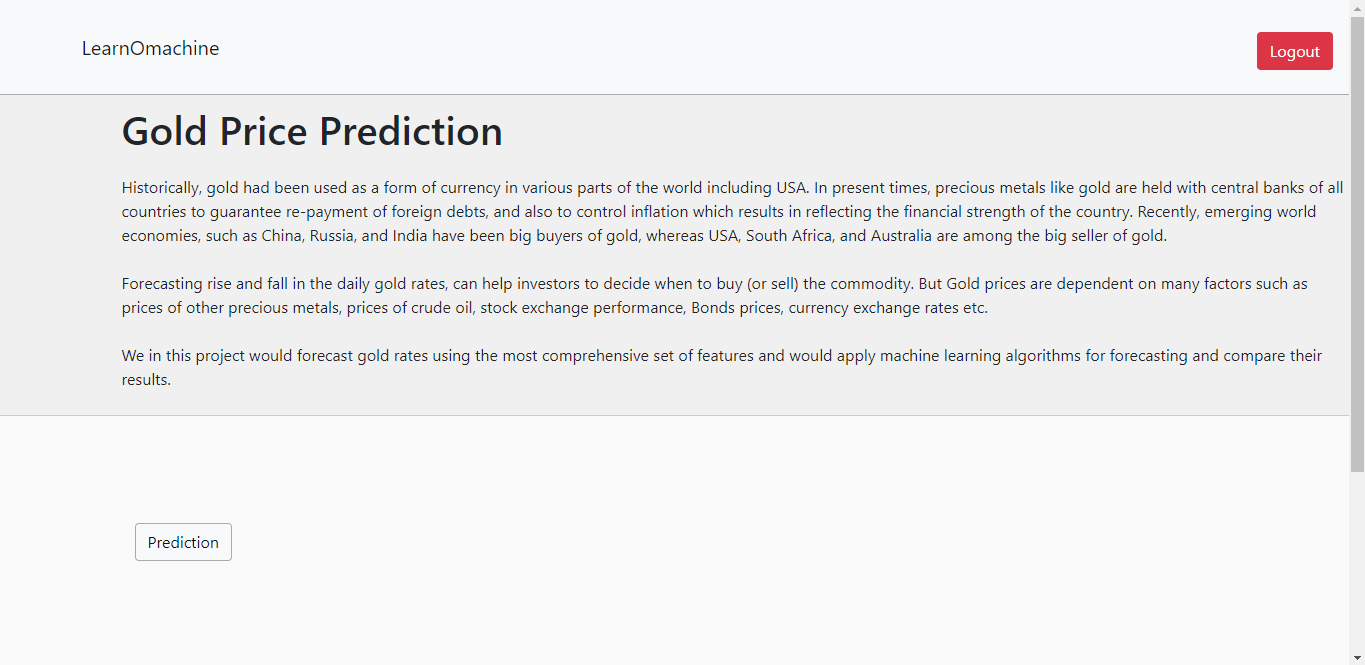
5.2 Work Breakdown Structure



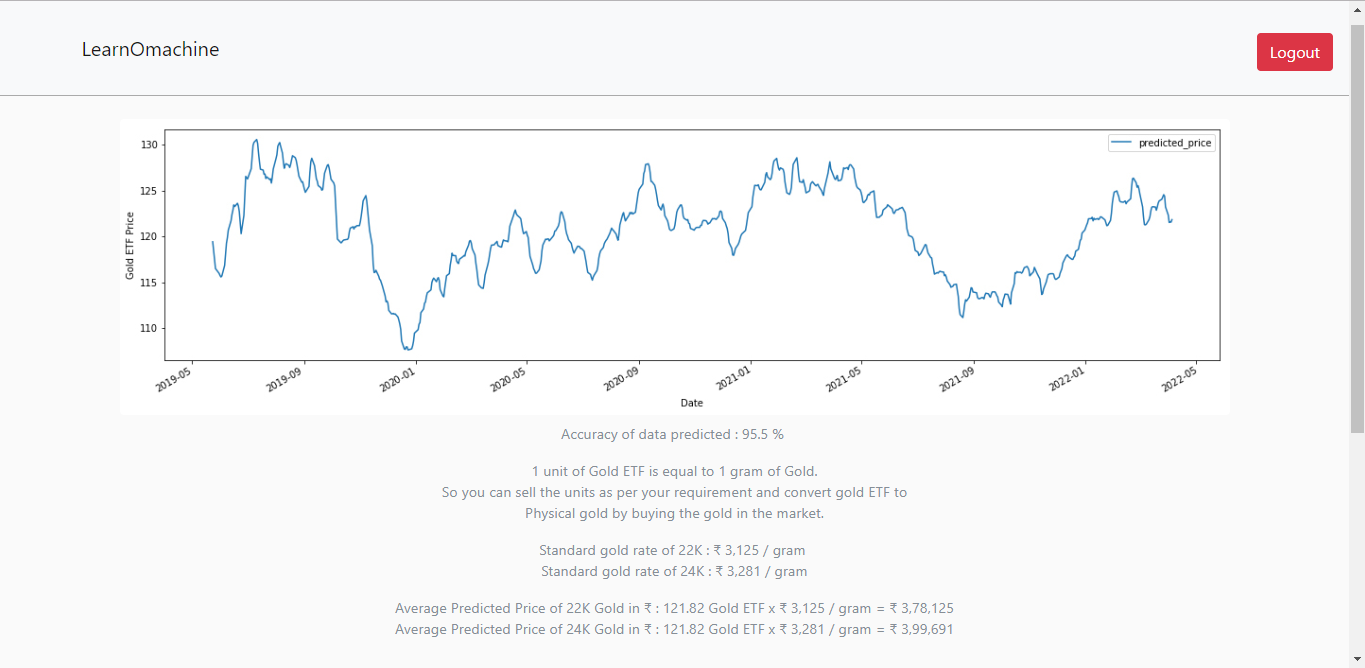
5.3 PERT Chart

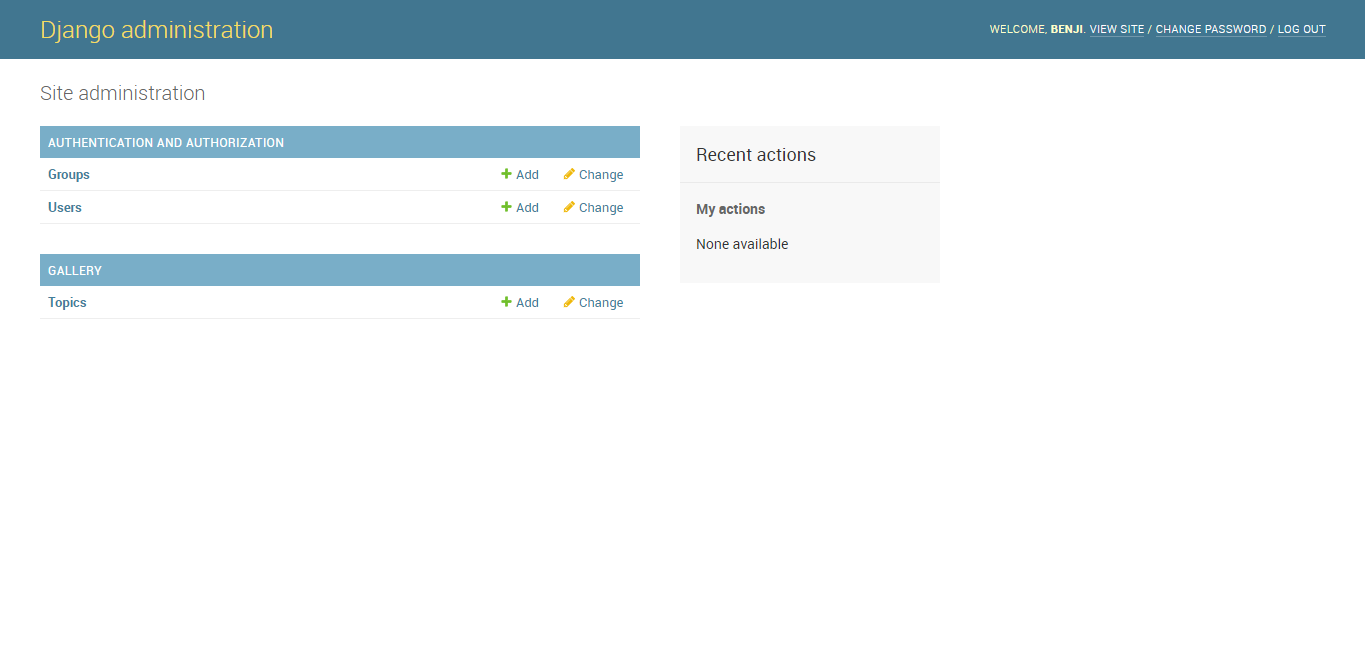
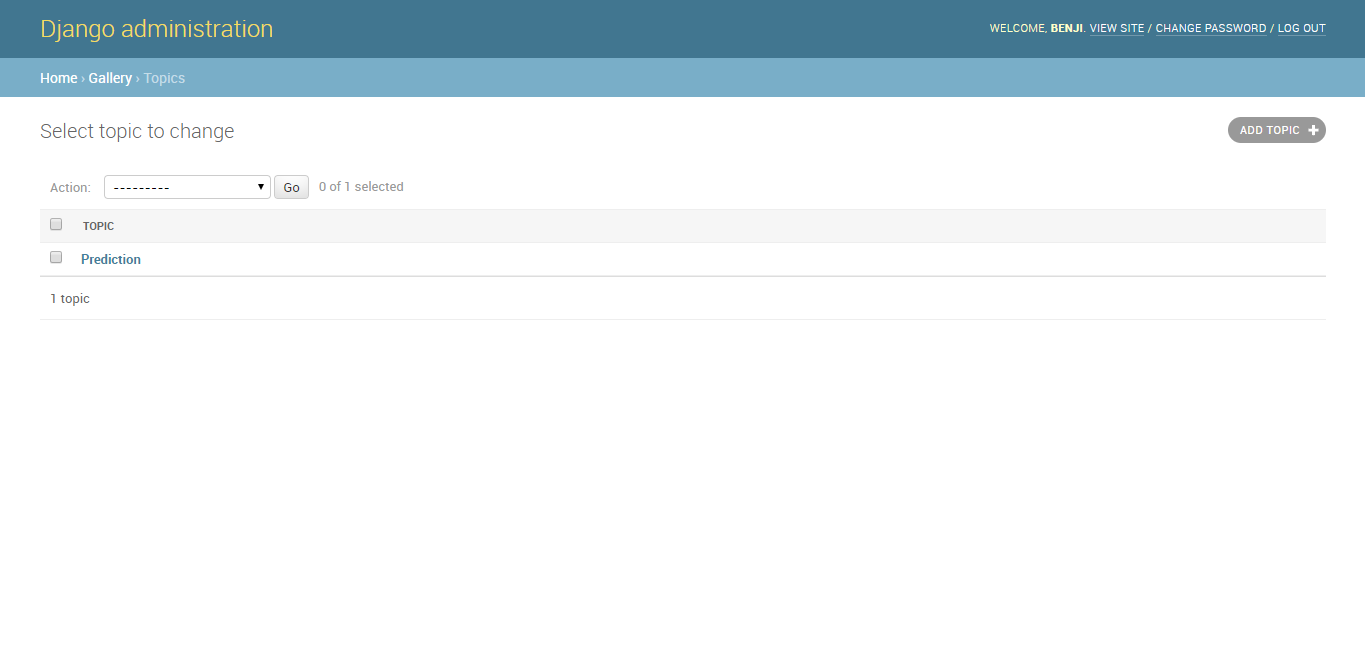


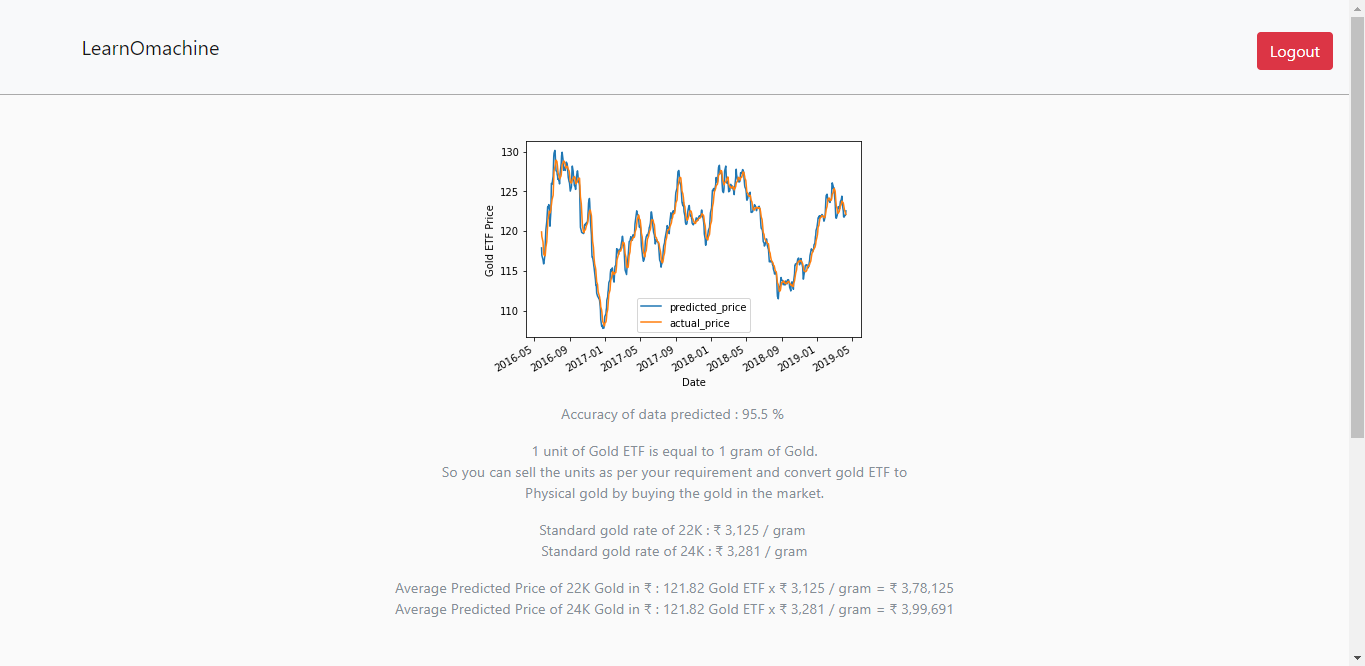
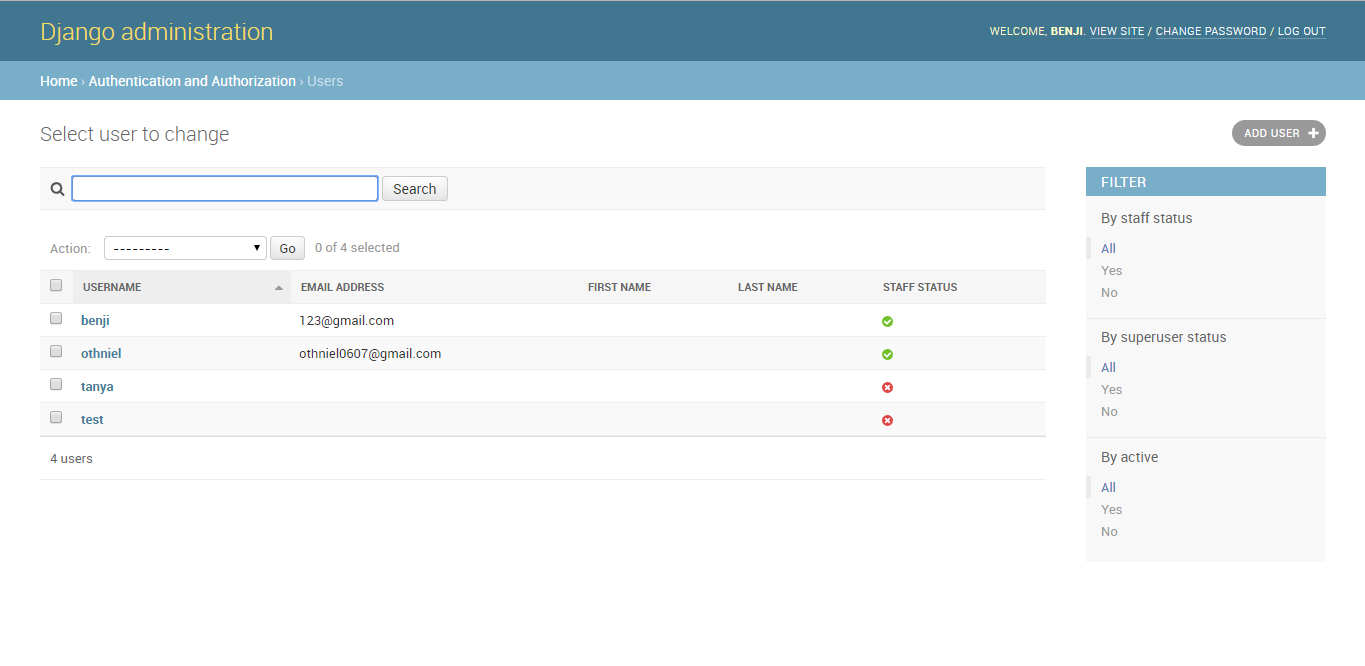
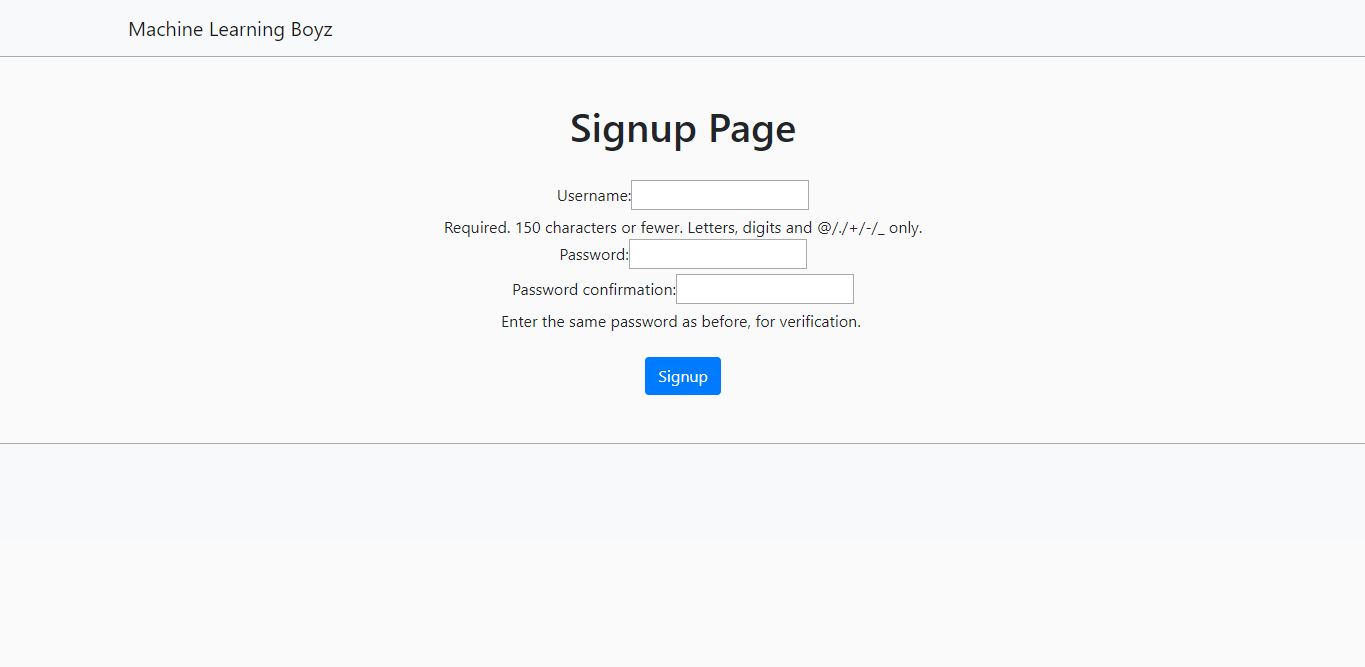
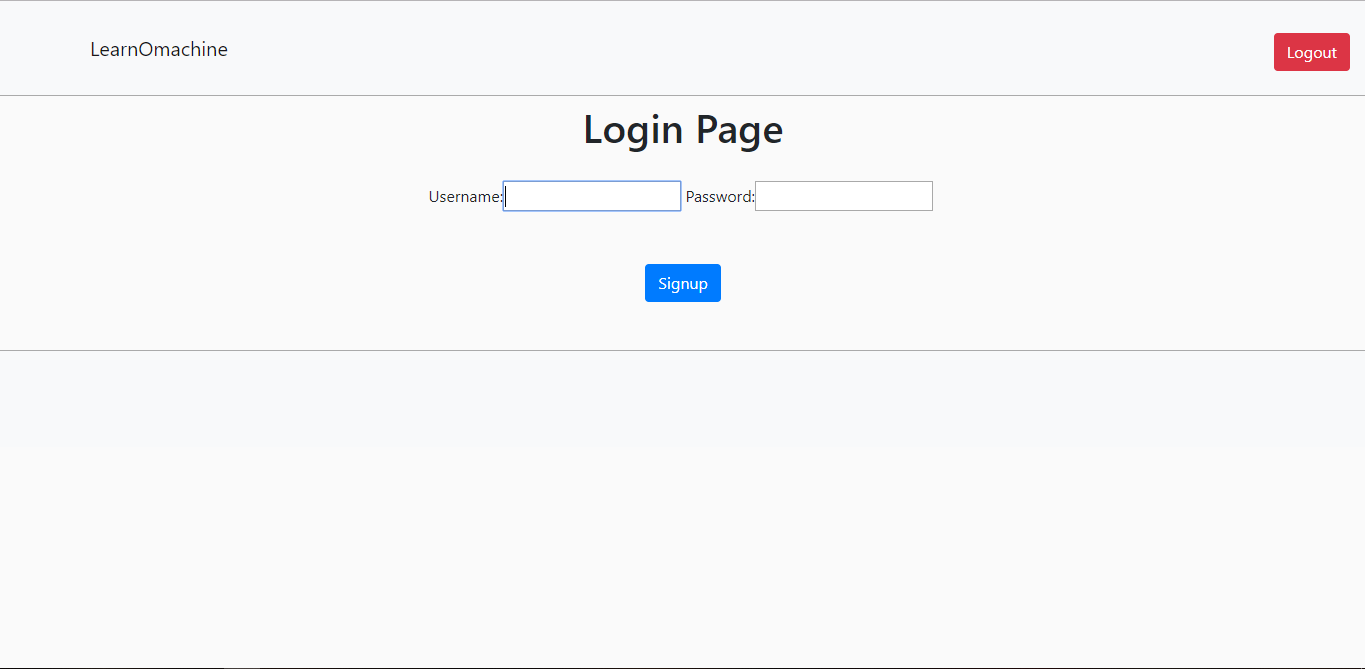
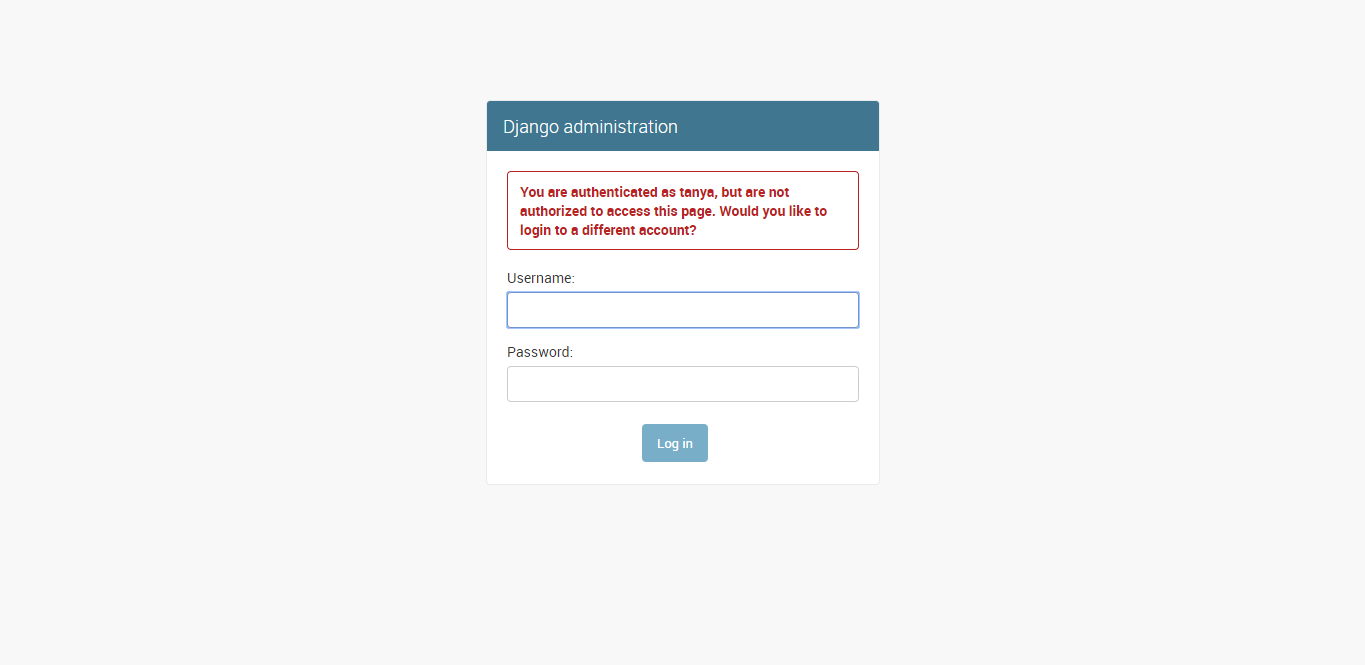
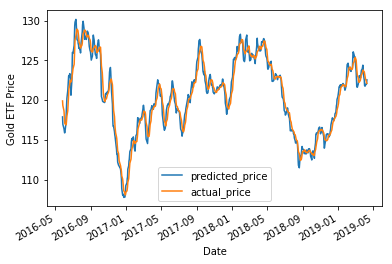
5.3 SNAPSHOTS OF UI











5.4 Test Cases

Table 5.4.1: Test Cases for Log-in

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Test** | **Test Case** | **Test Data** | | | **Expected Output** | | **Actual Output** | **Result** | | |
| **Case Id** | | **Name** |  | |  |  |  |  |  |  | |
|  |  |  |  | |  |  |  |  |  |  | |
| 1 | | User enters | Enters correct | | | Login Successfully | | Logged-in | Pass | | |
|  |  | username & | username & | | |  |  |  |  |  | |
|  |  | password | password- | | |  |  |  |  |  | |
|  |  |  | “12345” & “abcd” | | |  |  |  |  |  | |
|  |  |  |  | |  |  |  |  |  |  | |
| 2 | | User enters | Enters wrong | | | Prompt Error | | Prompt Error | Pass | | |
|  |  | username & | username & | | |  |  |  |  |  | |
|  |  | password | password- | | |  |  |  |  |  | |
|  |  |  | “123”,“xyz” | | |  |  |  |  |  | |
|  |  |  |  | |  |  |  |  |  |  | |
|  |  |  |  | |  | |  |  |  |  | |
|  |  |  |  | | Table 5.4.1: Test Cases for  Sign-up | |  |  |  |  | |
|  |  |  |  | | |  | |  |  | | |
|  | **Test** | **Test Case** | **Test Data** | | | **Expected Output** | | **Actual Output** | **Result** | | |
| **Case Id** | | **Name** |  | |  |  |  |  |  |  | |
|  | |  |  | | |  | |  |  | | |
| 1 | | Enter | Valid username & | | | Successfully | | Logged-in>Home | Pass | | |
|  |  | Username & | Password which | | | signed-up | | Page |  | |  |
|  |  | Password | doesn’t exist in | | |  |  |  |  | |  |
|  |  |  | database | | |  |  |  |  | |  |
|  | |  |  | | |  | |  |  | | |
| 2 | | Enter | Invalid username | | | Error | | Prompts Error | Pass | | |
|  |  | username & |  | and | |  |  |  |  | |  |
|  |  | password | password(that | | |  |  |  |  | |  |
|  |  |  | contains in | | |  |  |  |  | |  |
|  |  |  | database) | | |  |  |  |  | |  |
|  |  |  |  | |  |  |  |  |  | |  |

**6. Reflection**

The process undertaken this project are as follows :

*  Set Up Infrastructure
*  I Python Notebook
*  Incorporate required libraries Sklearn, Numpy, Pandas, Matplotlib and Seaborn

Data-set Preparation :

*  Incorporate data from different sources
*  Process the data into pandas data-frame
*  Normalize the data using Sklearn’s MinMax Function
*  Time-series Split with n\_splits=10
*  Develop Benchmark Model
* Set up basic decision tree regressor with default parameters as benchmark model
*  Develop Solution Model & improve using GridsearchCV
*  Evaluate all the solution models and benchmark model & Plot the result
*  Refine the models using Feature selection
*  Evaluate and compare the results with Original feature models
*  Plot, analyze and describe the results for report.

It is almost practically impossible to get a model that can 100% predict the price without any error, there are too many factors on which gold prices depends I tried to capture as much factors as possible but still there are other factors such as CPI, political factors, disasters, financial breakdown that can affect the market. But this solution model performed really well in this project, which will help the trader to make better decision. The general trend of predicted price is in line with the actual data, so the trader could have an indicator to reference, and makes trading decision by himself.

1. **Future Enhancements**

There are still many technical indicators and feature variables which we have not included in our project, may be there are some other indicators which we haven’t explored would perform better.

There are lots of Machine Learning algorithms which we haven’t tried and may be neural network or LSTM would perform better than our solution.

And last but not the least we have used data of around 14 years if we would increase the data, I think the performance of our solution models may be improved.

1. **References**
2. Bombay Stock Exchange
3. Yahoo Finance
4. Kaggle
5. Matplotlib, Pandas, SciKit Learn Documentation
6. Django tutorials (Net Ninja)
7. Machine Learning Tutorials (Youtube)