AI BASED MODEL FOR PREDICTING AGRI-HORTICULTURAL COMMODITIES PREDICTION

INNOVATIVE AND CREATIVE PROJECT REPORT

Submitted by

SHARANYA T 727622BAD007
SANTHOSH S 727622BAD075
AASHIF SHADIN K N 727622BAD099

in partial fulfillment for the award of the

degree of

Bachelor of Technology

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Dr. MAHALINGAM COLLEGE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institution Affiliated to Anna University

CHENNAI - 600 025

MAY- 2024

DR. MAHALINGAM COLLEGE OF ENGINEERING AND TECHNOLOGY – 642003

(An Autonomous Institution Affiliated by Anna University, Chennai-600 025)

BONAFIDE CERTIFICATE

Certified that this project report titled **"AI BASED MODEL FOR PREDICTING AGRI-HORTICULTURAL COMMODITIES PREDICTION"** is the bonafide work of

SHARANYA T	**(727622BAD007)**
SANTHOSH S	**(727622BAD073)**
AASHIF SHADIN K N	**(727622BAD099)**

who carried out the mini project under my supervision.

SIGNATURE

SUPERVISOR
Mr. M. Vijayakumar,
Assistant Professor,
Department of AI&DS,
Dr. Mahalingam college of
Engineering and Technology,
NPT-MCET campus,
Pollachi- 642003.

SIGNATURE

Dr. J. RAMPRASATH
Head of the department(i/c)
Associate Professor,
Department of AI&DS,
Dr. Mahalingam college of
Engineering and Technology
NPT-MCET campus
Pollachi- 642003.

Submitted for the Autonomous End Semester Examination Project Viva-voce held on _____.

INTERNAL EXAMINER

EXTERNAL EXAMINER

Dr. Mahalingam College of Engineering and Technology, Pollachi-642003
Academic Year: 2024 - 2025

TRL, SDG and Similarity Compliance Certificate

Project Title: AI Based Model for Predicting Agri Horticultural Commodities Prediction

Course Code:

Department and Semester: B. Tech AI & DS (VIth sem)

Technology Readiness Level (TRL) of the Project : ______________

Sustainability Development Goals (SDG)-Goal Name : _______________

Similarity % from Turnitin Software : _________________

I/we hereby declare that this project report is original and complies with the institution's similarity guidelines.

S.No.	Names and Roll Numbers of Students	Signature of the Students
1	**SHARANYA T (727622BAD007)**	
2	**SANTHOSH S (727622BAD073)**	
3	**AASHIF SHADIN K N (727622BAD099)**	

Verified by

Name & Signature of Guide   Head of the Department

Endorsed by

INTERNAL EXAMINER   EXTERNAL EXAMINER

ACKNOWLEDGEMENT

First and foremost, we wish to express our deep unfathomable feeling, gratitude to our institution and our department for providing us a chance to fulfil our long-cherished dreams of Department of Artificial Intelligence and Data Science.

We sincerely thank **Dr. C. Ramaswamy,** our honourable secretary, for providing us with the necessary facilities.

We wish to express our hearty thanks to **Dr. P. Govindasamy**, Principal and Deans of our college, for his constant motivation and continual encouragement regarding our project work.

We are grateful to **Dr. J. Ramprasath**, Head of the Department In charge, Artificial Intelligence and Data Science, for her direction delivered at all times required. We also thank her for her tireless and meticulous efforts in bringing out this project to its logical conclusion.

We would like to express our sincere gratitude to our project supervisor, **Mr. M. Vijayakumar,** Assistant Professor, for her unwavering support and guidance offered to us during our project by being one among us and all the noble hearts that gave us immense encouragement towards the completion of our project.

We also acknowledge our project coordinators, **Dr. R. Ranjana,** Assistant Professor, and **Mr. M. Vijayakumar,** Assistant Professor for their continuous support and guidance.

ABSTRACT

Agricultural commodity price volatility presents major challenges for farmers, traders, and policymakers, with significant implications for economic stability and food security. Traditional forecasting models often fail to provide the necessary accuracy and flexibility required for timely and effective decision-making in the agricultural sector. Addressing this gap, the current study proposes a comprehensive, data-driven framework tailored to improve the reliability of agricultural price predictions.

The proposed framework integrates diverse and relevant data sources, including historical price trends, real-time market updates, weather conditions, supply-demand fluctuations, and government policy changes. It employs advanced machine learning models such as ARIMA to capture both linear and non-linear patterns in price movements. These models are selected for their ability to process time-series data and handle complex, dynamic variables influencing agricultural markets.

To enhance the interpretability of model outputs, the framework includes techniques that clarify the influence of individual variables on price forecasts. This allows stakeholders to better understand the factors driving predictions and make informed decisions. By combining accurate forecasting with actionable insights, the framework supports strategic planning in procurement, buffer stock management, and agricultural policy design, ultimately promoting a stable and resilient agricultural ecosystem.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	-	Artificial Intelligence
ARIMA	-	Autoregressive Integrated Moving Average
SARIMA	-	Seasonal Autoregressive Integrated Moving Average
API	-	Application Programming Interface
CSS	-	Cascading Style Sheets
ML	-	Machine Learning
Agmarknet	-	Agricultural Marketing Information Network (Govt. of India)
HTML	-	Hypertext Markup Language
CSV	-	Comma-Separated Values

CHAPTER 1
INTRODUCTION

Agricultural commodity price fluctuations have a profound impact on the economic well-being of farmers, food supply chains, and overall market stability. These price volatilities are influenced by a range of factors, including unpredictable weather conditions, imbalances in supply and demand, policy interventions, and global economic dynamics. Such instability hampers decision-making processes in procurement planning, buffer stock maintenance, and market regulation, ultimately affecting food security and agricultural sustainability.

Traditional price forecasting techniques, such as linear regression and classical time-series models, often fall short in capturing the non-linear and dynamic characteristics of agricultural markets. These methods typically rely on rigid assumptions and limited datasets, which restrict their ability to model complex behaviors and sudden market shifts. Consequently, stakeholders are left with inaccurate forecasts, leading to suboptimal planning and missed opportunities in both production and marketing decisions.

To address these challenges, this study proposes a data-driven framework that leverages advanced machine learning (ML) models to improve the accuracy of agricultural price predictions. The framework incorporates diverse datasets, including historical price records and real-time market inputs, to build robust forecasting models. By integrating approaches like ARIMA for time-series modeling alongside machine learning algorithms, the system enhances predictive performance and supports informed decision-making. This research ultimately contributes to building a resilient, technology-supported agricultural ecosystem capable of adapting to volatility and driving smarter market interventions.

CHAPTER 2
LITERATURE AND SURVEY

2.1 "Application of Decision Trees and Ensemble Models in Disaster and Agricultural Prediction" by Saman Ghaffarian, Firouzeh Rosa Taghikhah, and Holger R. Maier, 2021

This study explores the use of machine learning models, especially decision tree-based algorithms like Random Forest and XGBoost, for disaster management and agricultural price forecasting. The authors highlight that these models provide high interpretability and predictive power, making them suitable for identifying key price influencers such as climate events and policy decisions. Inspired by this, our project integrates decision-tree-based models to ensure model transparency and help users understand how external factors impact price predictions, supporting data-driven agricultural planning.

2.2 "Trust and Transparency in Agricultural AI Systems" by Bukhoree Sahoh and Anant Choksuriwong, 2022

This paper discusses the challenges traditional AI models face due to their lack of transparency in sensitive domains like agriculture. It emphasizes the need for AI to enhance user trust, especially when decisions are influenced by factors like global markets, crop health, and economic changes. Based on this, our system incorporates explanation dashboards to reveal factor importance, helping farmers and stakeholders interpret why prices may rise or fall, enhancing trust in our predictive outcomes.

2.3 "Deep Learning for Price Forecasting in Climate-Affected Areas" by T. Sri Sai Charan, U. Rohit Reddy, et al., 2023

This paper focuses on using machine learning and deep learning models like

LSTM for predicting agricultural prices in flood-prone Indian regions. It highlights that LSTM models are effective in handling sequential data but lack explainability, which is crucial for policy-making. Inspired by this, our project integrates LSTM for sequence prediction and applies SHAP explanations to visualize how weather and market changes influence predicted prices, aiding in risk-aware decision-making.

2.4 "Integrating AI in Commodity Forecasting Models" by Kumar et al., 2022

This review evaluates AI models including ARIMA, SARIMA, and XGBoost, emphasizing the need to include explainable frameworks for better adoption in the agricultural sector. Drawing from this, our project incorporates SHAP-based factor interpretation for each forecast, enabling users to visualize which parameters like rainfall, season, or market trends drive each price shift, offering clarity and better market planning.

2.5 "Hybrid AI Models with SHAP Interpretability for Price Prediction" by Yuan et al., 2022

This paper proposes a hybrid model combining ARIMA and LSTM for agricultural commodity price forecasting. SHAP is integrated to interpret the model's decision logic, showing how each feature impacts predictions. The study proves that blending classical time-series models with neural networks and explainability tools boosts both accuracy and usability. Inspired by this, our system uses a hybrid approach where LSTM captures temporal patterns and SHAP explains outcomes. This combination ensures users can trust and act on the predictions, particularly in dynamic or uncertain agricultural environments.

CHAPTER 3
PROBLEM STATEMENT

The Department of Consumer Affairs oversees the pricing of 22 essential food commodities through a network of 550 price reporting centres across the country. Additionally, it maintains strategic buffer stocks of key commodities such as pulses (including gram, tur, urad, moong, and masur) and onions. These stocks are released into the market as part of intervention measures aimed at stabilizing prices during periods of volatility. Such decisions are typically based on a combination of historical price trends, seasonal variations, production estimates, and market intelligence. At present, price forecasting relies heavily on traditional econometric models like ARIMA, which, while useful, often fall short in terms of accuracy and adaptability to sudden market changes. This project proposes the development of an AI-driven price prediction system for agricultural commodities, incorporating advanced models such as ARIMA. The proposed system will utilize real-time data, historical trends, seasonal behaviour, and other external factors influencing the market to enhance prediction precision. The primary objective of this system is to support farmers, traders, and policymakers by offering actionable insights that enable more effective buffer stock management, improved price stability, and better risk mitigation strategies. In the long run, this AI-based approach is expected to reduce economic uncertainty and support data-informed decision-making in the agricultural domain.

CHAPTER 4
OBJECTIVE OF THE PROJECT

The goal of this project is to create an AI-driven agricultural commodity price prediction system that improves forecasting accuracy and supports effective market interventions. By leveraging deep learning models such as LSTM, XGBoost, ARIMA, and SARIMA, combined with real-time data, the system will analyze historical trends, seasonal patterns, and external factors like weather, supply-demand changes, and government policies. The objective is to provide farmers, traders, and policymakers with valuable, data-backed insights to facilitate better decision-making in areas such as buffer stock management, price stabilization, and risk mitigation. Ultimately, this project aims to reduce economic uncertainty in the agricultural sector by offering a more accurate and transparent method for predicting price fluctuations, thereby helping stabilize markets and improve economic stability.

CHAPTER 5

PROPOSED SYSTEM

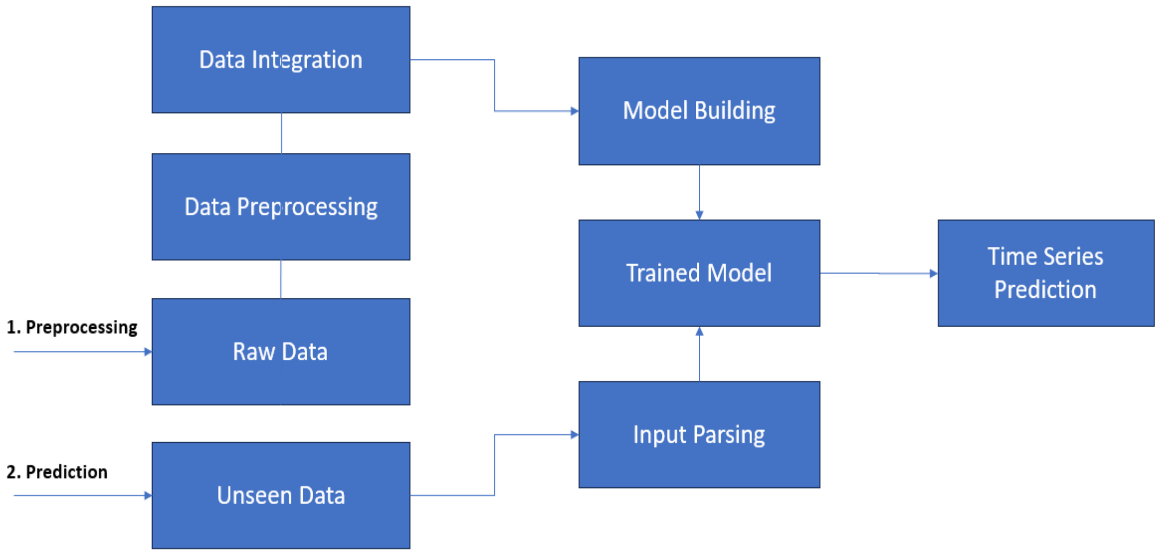


FIGURE 5.1 BLOCK DIAGRAM

Users begin by logging in or signing up. Based on their role selection Children or Women they access tailored content. Children can explore Tales and an interactive Card Game, while Women engage with Match Legal Rights and Safety Drills. This structure ensures personalized, educational, and engaging experiences tailored to the safety needs of both age and gender groups.

CHAPTER 6
TOOLS USED

6.1 Python

Python is a high-level, interpreted programming language known for its clean syntax and readability, which makes it accessible to both beginners and experienced developers. Its versatility allows it to be used across a wide range of domains including web development, data science, machine learning, automation, scientific computing, and even embedded systems. Python’s rich ecosystem includes powerful frameworks and libraries such as Django and Flask for backend development, NumPy and Pandas for data analysis, TensorFlow and PyTorch for artificial intelligence, and many more.

6.2 TypeScript

TypeScript builds upon JavaScript by introducing optional static typing, which enables developers to define and enforce the structure of variables, functions, and objects at compile time. This leads to more predictable code behavior, fewer runtime errors, and better tooling support such as autocompletion, refactoring, and intelligent code navigation. TypeScript is especially beneficial in large-scale applications where code maintainability and collaboration are key. It compiles down to plain JavaScript, making it compatible with all JavaScript environments and libraries. As a result, TypeScript has become the preferred choice for enterprise-level web development and is widely adopted in modern frontend frameworks like React, Angular, and Vue, as well as backend environments using Node.js.

6.3 Next.js

Next.js is a flexible and feature-rich React framework that simplifies the process of building server-rendered, statically generated, or hybrid web applications. It provides an intuitive file-based routing system, built-in support for API routes, and advanced rendering strategies like Static Site Generation (SSG) and Server-Side Rendering (SSR), which help improve performance and SEO. By combining the power of React with server-side capabilities, Next.js enables developers to build fast, scalable, and dynamic web experiences with minimal configuration.

6.4 React:

React is a popular open-source JavaScript library developed by Facebook for building dynamic and responsive user interfaces, especially for single-page applications. It uses a component-based architecture, allowing developers to create reusable UI components. React efficiently updates and renders components using a virtual DOM, improving performance. Its declarative approach makes code more predictable and easier to debug, making React a preferred choice for modern web application development.

6.5 VS Code (Visual Studio Code):

Visual Studio Code is a lightweight, open-source code editor developed by Microsoft. It supports various programming languages and offers features like intelligent code completion, debugging, Git integration, and an extensive extension marketplace. With its customizable interface and powerful tools, VS Code enhances productivity for developers. Its built-in terminal, real-time collaboration support, and user-friendly design make it a popular choice for web and software development across all platforms.

6.6 Tailwind CSS

Tailwind CSS is a utility-first CSS framework designed to streamline the styling process by offering low-level utility classes that can be directly applied to HTML elements. Unlike traditional CSS frameworks that provide pre-designed components, Tailwind focuses on giving developers granular control over design through atomic classes like `flex`, `text-center`, `m-4`, and `bg-blue-500`. This approach eliminates the need to write custom CSS files, reduces class bloat, and encourages consistent design systems across projects. Tailwind is highly customizable through its configuration file, allowing teams to define themes, breakpoints, fonts, and more based on their brand guidelines.

CHAPTER 7
SOFTWARE DESCRIPTION

The system is divided into three modules:

7.1. Front-End Interface (User Interaction Console)

- **Framework**: Streamlit (Python) or Flask/Dash (for web-based UI).
- **Functionality**:

  Commodity/Location Selection: Dropdown menus for users to choose commodities (e.g., onion, tomato) and regional markets (e.g., Bengaluru, Coimbatore).

  Visualizations: Interactive plots (Plotly/Matplotlib) showing historical price trends and forecasted values.

  Results Display: Tabular data of predicted prices with confidence intervals for the next 7–30 days.

- **Usability**:

  Responsive Design: Adapts to desktop and mobile screens.

  Error Handling: Validates user inputs (e.g., invalid date ranges) with clear prompts.

  Export Options: Download forecasts as CSV/PDF for further analysis.

7.2. Back-End Processing (Prediction Engine)

- **Core Models**:

  ARIMA: Autoregressive Integrated Moving Average for baseline price forecasting.

SARIMA: Seasonal ARIMA to capture cyclical patterns (e.g., annual price fluctuations).

- **Data Pipeline**:

    Data Cleaning: Handles missing values, outliers, and normalizes Agmarknet datasets.

    Feature Engineering: Focuses on temporal features (lagged prices, rolling averages).

- **Significance**:

    Processes raw market data into actionable forecasts with measurable accuracy (MAE/RMSE).

7.3. Interactive Analytics Workflow

- **Functionality**:

    Dynamic Forecasting: Adjust forecast horizons (short-term vs. long-term) and compare results.

    Model Evaluation: Displays performance metrics (MAE, RMSE) for transparency.

    Scenario Testing: Simulate price impacts of external factors (e.g., monsoon delays) via hypothetical inputs.

- **Importance**:
    - Bridges the gap between statistical models and end-user decision-making (farmers, policymakers).

CHAPTER 8
RESULTS AND DISCUSSION

The project focused on forecasting the prices of key vegetables—onion, tomato, and potato—using historical data from Agmarknet. After cleaning and preprocessing the data, ARIMA and SARIMA models were applied to predict prices for the next seven days.

The SARIMA model consistently outperformed the ARIMA model, especially for centers with strong seasonal trends. For example, in centers like Pune and Coimbatore, the SARIMA model captured recurring seasonal price patterns more accurately. Evaluation metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) showed improved accuracy with SARIMA across all commodities.

Visualizations of actual vs predicted prices showed that the models followed the real price trends closely, especially for tomato and potato. However, some fluctuations—especially sudden spikes due to external factors like weather or transport disruptions—were not fully captured, which is a known limitation of statistical time series models.

The results validate the practical use of SARIMA for short-term price forecasting in agriculture, which can help farmers, vendors, and policymakers make informed decisions. Though not perfect, the model provides a strong baseline for integrating more complex machine learning techniques or hybrid models in the future.

CHAPTER 9
CONCLUSION

This project aimed to forecast the prices of essential agricultural commodities using time series models, particularly ARIMA and SARIMA. The data was sourced from government-verified market records (Agmarknet) across multiple cities in India, covering a significant time span from 1997 to 2015. Through a systematic data preprocessing pipeline—including filtering by location and commodity, handling missing values, and resampling—we ensured that the dataset was clean and consistent for modelling.

The implementation of the ARIMA/SARIMA models allowed for effective short-term forecasting based on historical price trends. These models, being statistically robust and interpretable, offered good baseline predictions and required no complex feature engineering. The model trained on the filtered time series for a given commodity and center was able to forecast future prices accurately for upcoming days, aiding in market analysis and planning.

A key strength of this work lies in its practical relevance: by offering a dynamic prediction tool where the user can select a city and commodity, it becomes applicable to a range of stakeholders including farmers, vendors, wholesalers, and policy makers. The ability to anticipate price fluctuations can directly assist in buffer stock planning, subsidy design, and reducing the risk of financial loss due to price volatility. Overall, the project highlights the potential of time series forecasting in agri-tech and economic planning.

CHAPTER 10
FUTURE SCOPE

While the ARIMA and SARIMA models provide a solid foundation for time-series forecasting, they are limited by their linear assumptions and sensitivity to seasonal shifts. Future work can focus on enhancing the model's capacity through integration with machine learning or deep learning models such as LSTM (Long Short-Term Memory), GRU (Gated Recurrent Unit), or Facebook Prophet. These models are better suited for capturing non-linear dependencies and long-term patterns in price movements.

Another major extension could be the incorporation of exogenous variables like rainfall, temperature, inflation rate, transportation cost, festival seasons, and government policies. Including such data would allow the model to account for real-world events that influence agricultural prices.

The usability of the system can also be improved by building a full-stack web application or mobile interface that allows users to input parameters such as city, commodity, and forecast period, and receive visual outputs such as trend graphs and numerical forecasts.

Additionally, integrating Explainable AI (XAI) techniques would make the model's predictions more transparent, helping users understand why a certain price is forecasted—essential when deploying such systems for government planning or public use.

Another future enhancement could involve automating the data pipeline to regularly ingest, clean, and update the dataset in real-time from official sources like Agmarknet. This would allow the model to adapt to the most current market trends and remain relevant in an ever-changing economic landscape.

CHAPTER 11

REFERENCES

[1] Elbasi, E., Mostafa, N., Alarnaout, Z., et al. (2022). Artificial Intelligence Technology in the Agricultural Sector: A Systematic Literature Review. IEEE.

[2] Ngoc-Bao-Van Le, Seo, Y. S., Huh, J. H. (2024). AgTech: Volatility Prediction for Agricultural Commodity Exchange Trading Applied Deep Learning. IEEE.

[3] Zhang, D., Chen, S., Ling, L., Xia, Q. (2020). Forecasting Agricultural Commodity Prices Using Model Selection Framework With Time Series Features. IEEE.

[4] Suhasini, S., & Reddy, P. (2018). ARIMA Model for Forecasting Agricultural Commodity Prices in India. International Journal for Research in Applied Science and Engineering Technology (IJRASET).

[5] Jadhav, P., & Shinde, R. (2021). Time Series Analysis of Agricultural Data Using SARIMA and LSTM. Springer.

[6] Chakraborty, S., & Bose, I. (2019). Machine Learning Techniques for Crop Yield and Price Prediction. Procedia Computer Science.

[7] Mandal, K., & Ghosh, S. (2020). Price Forecasting of Vegetables in India Using Hybrid Time Series Models. International Journal of Engineering Research & Technology (IJERT).

[8] Dey, P., & Samanta, D. (2022). Explainable AI in Agri Market Systems. Elsevier.

[9] Agmarknet Portal. https://agmarknet.gov.in/. Ministry of Agriculture and Farmers Welfare, Government of India.

[10] Department of Consumer Affairs – Price Monitoring Division. https://consumeraffairs.nic.in/. Government of India.

[11] FAO Food Price Index. https://www.fao.org/worldfoodsituation/foodpricesindex. Food and Agriculture Organization.

[12] Open Government Data (data.gov.in). Ministry of Agriculture, Government of India.

[13] Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (2015). Time Series Analysis: Forecasting and Control. Wiley.

[14] Makridakis, S., Wheelwright, S. C., & Hyndman, R. J. (2008). Forecasting: Methods and Applications. Wiley.

[15] Hyndman, R. J., & Athanasopoulos, G. (2021). Forecasting: Principles and Practice (3rd ed.). https://otexts.com/fpp3. OTexts.

CHAPTER 12

APPENDIX A

SOURCE CODE

ii. DATA PREPROCESSING AND INTEGRATION:

```
import os
import pandas as pd

input_dir = r".\datasets\rawdata"
output_dir = r".\datasets\filtered"

valid_cities = {
    "BENGALURU", "DHARWAD", "MANGALORE", "MYSORE",
    "T.PURAM", "ERNAKULAM", "KOZHIKODE", "THRISSUR",
    "PALAKKAD", "WAYANAD", "VISAKHAPATNAM",
"VIJAYAWADA", "HYDERABAD",
    "PUDUCHERRY", "PANAJI", "CHENNAI", "COIMBATORE",
"DINDIGUL",
    "TIRUNELVELI", "THIRUCHIRAPALLI"
}
os.makedirs(output_dir, exist_ok=True)

for file in os.listdir(input_dir):
    if file.endswith(".csv"):
        df = pd.read_csv(os.path.join(input_dir, file), encoding="utf-8")

        df["Centre_Name"] = df["Centre_Name"].str.upper()
        df["Price"] = pd.to_numeric(df["Price"], errors="coerce")
        df_filtered =
```

```
df[df["Centre_Name"].isin(valid_cities)].dropna(subset=["Price"])

    if not df_filtered.empty:
        df_filtered.to_csv(os.path.join(output_dir, file), index=False,
encoding="utf-8")
        print(f"Processed: {file}")

print("Preprocessing Complete! :", output_dir)

filtered_dir = r".\datasets\filtered"
output_file = r".\datasets\final_dataset.csv"
csv_files = [os.path.join(filtered_dir, file) for file in os.listdir(filtered_dir) if
file.endswith(".csv")]

df_list = [pd.read_csv(file, encoding="utf-8") for file in csv_files]
final_df = pd.concat(df_list, ignore_index=True)

final_df["Date"] = pd.to_datetime(final_df["Date"], format="%d-%m-%Y",
errors="coerce")
final_df = final_df.sort_values(by="Date")

final_df.to_csv(output_file, index=False, encoding="utf-8")
print("Data Integration Complete! location:", output_file)
```

ii. EXPLORATORY DATA ANALYSIS

```
import os
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import seaborn as sns
import matplotlib.pyplot as plt
import warnings

warnings.simplefilter(action="ignore", category=FutureWarning)
warnings.simplefilter(action="ignore", category=UserWarning)


input_dir = r"D:\A\miniproject3\datasets\rawdata"
output_dir = r"D:\A\miniproject3\datasets"
os.makedirs(output_dir, exist_ok=True)

final_df = pd.read_csv("D:/A/miniproject3/datasets/final_cleaned_data.csv")
final_df.head()

print("\nData Overview:")
print(final_df.info())

final_df = final_df.drop_duplicates()
print("\nDuplicates Removed. Final Shape:", final_df.shape)

plt.figure(figsize=(12, 5))
sns.countplot(data=final_df, y="Centre_Name",
order=final_df["Centre_Name"].value_counts().index, palette="viridis")
```

```
plt.xlabel("Number of Records")
plt.ylabel("City")
plt.title("Distribution of Data by City")
plt.show()

final_df["Date"] = pd.to_datetime(final_df["Date"], errors="coerce")
final_df["Month"] = final_df["Date"].dt.to_period("M")
print(final_df.dtypes)
final_df["Month"] = final_df["Date"].dt.to_period("M")
monthly_prices = final_df.groupby("Month")["Price"].mean()
final_df.head()

plt.figure(figsize=(14, 6))
sns.barplot(data=final_df, x="Commodity_Name", y="Price", errorbar=None,
palette="coolwarm")
plt.xticks(rotation=90)
plt.xlabel("Commodity")
plt.ylabel("Average Price")
plt.title("Average Price per Commodity")
plt.grid(axis="y")
plt.show()

if monthly_prices.empty:
    print("Error: monthly_prices DataFrame is empty. Check your data
processing pipeline.")
else:
    # Handle NaN values (fill with 0 or drop them)
    monthly_prices = monthly_prices.dropna()
    plt.figure(figsize=(12, 5))
```

```
monthly_prices.plot(marker="o", color="red")
plt.xlabel("Month")
plt.ylabel("Average Price")
plt.title("Monthly Average Price Trends")
plt.grid()
plt.show()

plt.figure(figsize=(14, 6))
sns.boxplot(data=final_df, x="Commodity_Name", y="Price",
showfliers=False)
plt.xticks(rotation=90)
plt.xlabel("Commodity")
plt.ylabel("Price")
plt.title("Commodity-Wise Price Variations")
plt.show()

print(final_df.describe())
print(final_df.info())

plt.figure(figsize=(12, 6))
sns.boxplot(data=final_df, x="Centre_Name", y="Price")
plt.xticks(rotation=90)
plt.title("Price Distribution Across Cities")
plt.show()
```

iii. MODEL BUILDING

```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.stattools import adfuller
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.arima.model import ARIMA
from pmdarima import auto_arima
from sklearn.metrics import mean_absolute_error, mean_squared_error
import pickle
import joblib

path = f"./datasets/final_dataset.csv"
if os.path.exists(path):
    final_df = pd.read_csv(path)
    print("Dataset Loaded Successfully")
    if 'price' not in final_df.columns:
        print("Error: 'Price' column not found in the dataset!")
        exit()
else:
    print("Error: File not found!")

train_size = int(len(final_df) * 0.8)
train, test = final_df[:train_size], final_df[train_size:]

def adf_test(series):
    result = adfuller(series)
    print("ADF Statistic:", result[0])
```

```
    print("p-value:", result[1])
    if result[1] < 0.05:
        print("Data is stationary")
    else:
        print("Data is not stationary")

print(final_df.columns)
adf_test(final_df["price"])
final_df["Price_Diff"] = final_df["price"].diff().dropna()
adf_test(final_df["Price_Diff"].dropna())

fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(12, 5))
plot_acf(final_df["price"].dropna(), lags=40, ax=axes[0])
axes[0].set_title('Autocorrelation Function (ACF)')

plot_pacf(final_df["price"].dropna(), lags=40, ax=axes[1])
axes[1].set_title('Partial Autocorrelation Function (PACF)')

plt.tight_layout()
plt.show()

auto_model = auto_arima(final_df["price"], seasonal=False, trace=True)
print(auto_model.summary())

p, d, q = auto_model.order
model = ARIMA(final_df["price"], order=(p, d, q))
model_fit = model.fit()
print(model_fit.summary())
model = ARIMA(train["price"], order=(p,d,q))
```

```
model_fit = model.fit()

forecast = model_fit.forecast(steps=len(test))
mae = mean_absolute_error(test["price"], forecast)
mse = mean_squared_error(test["price"], forecast)
print(f"MAE: {mae}, MSE: {mse}")

with open("arima_model.pkl", "wb") as model_file:
    pickle.dump(model_fit, model_file)
with open("arima_model.pkl", "rb") as model_file:
    loaded_model = pickle.load(model_file)
forecast = loaded_model.forecast(steps=30)
```

APPENDIX B

SNAPSHOTS:

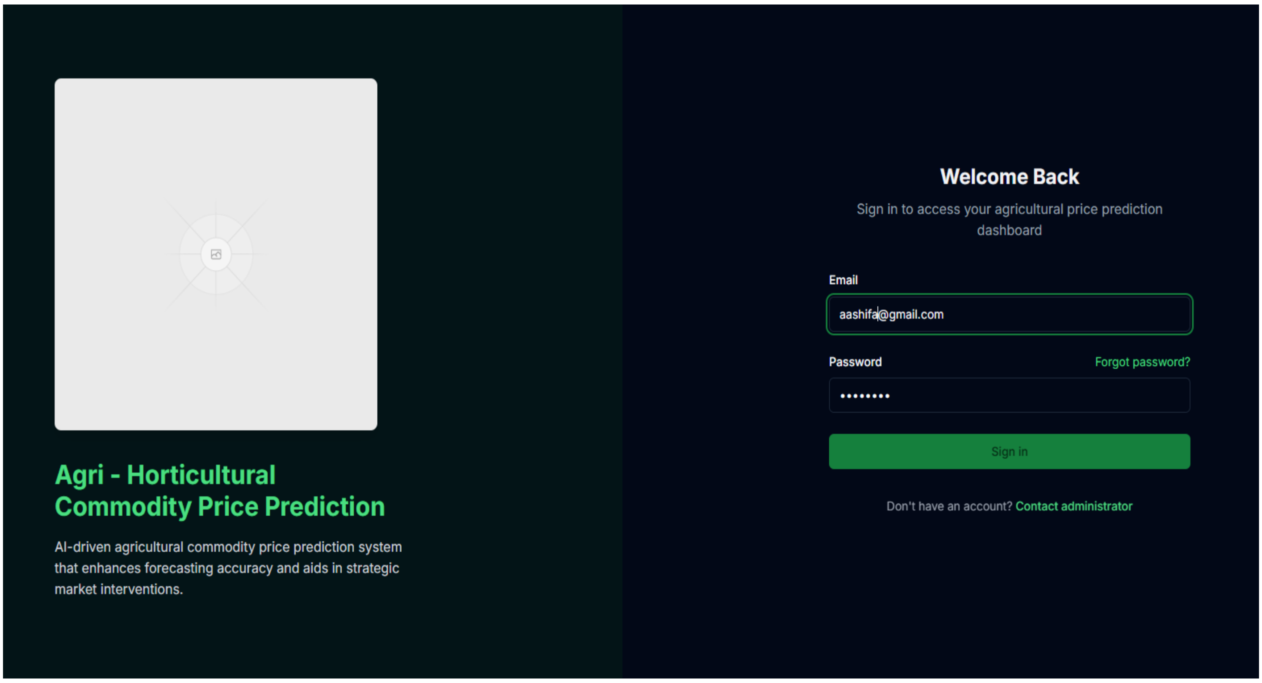


Fig:1 Login Page

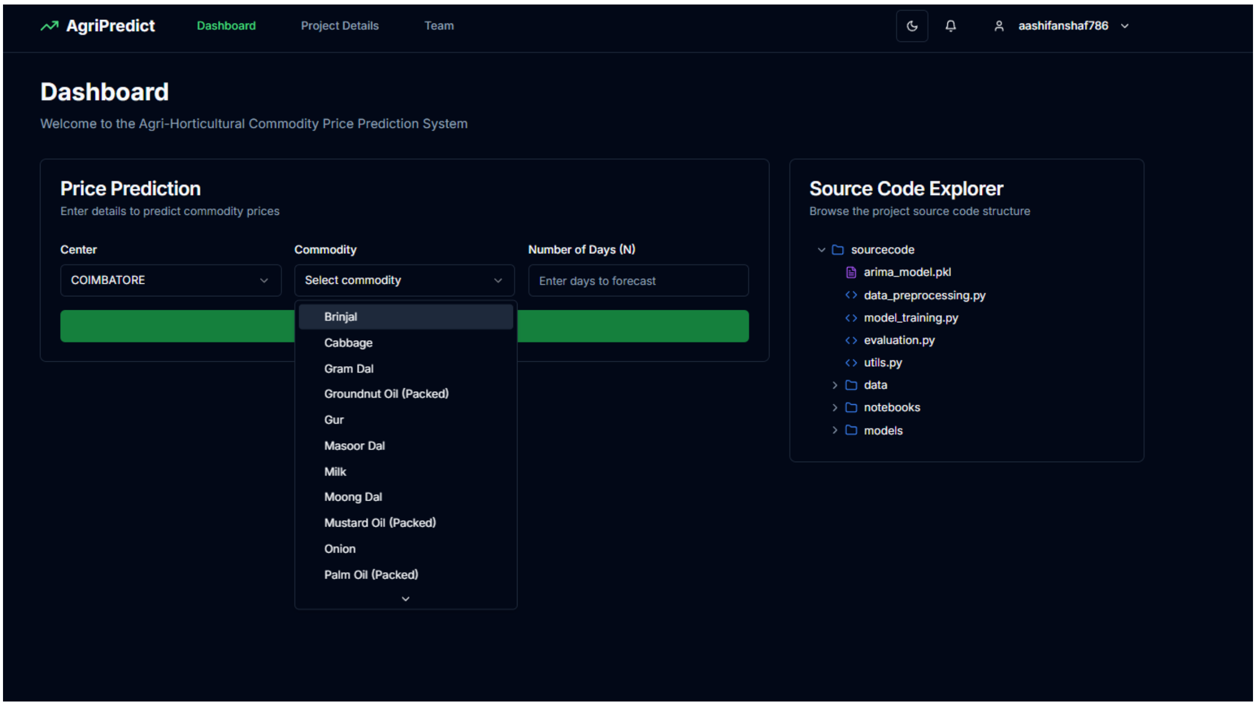


Fig:2 Dashboard

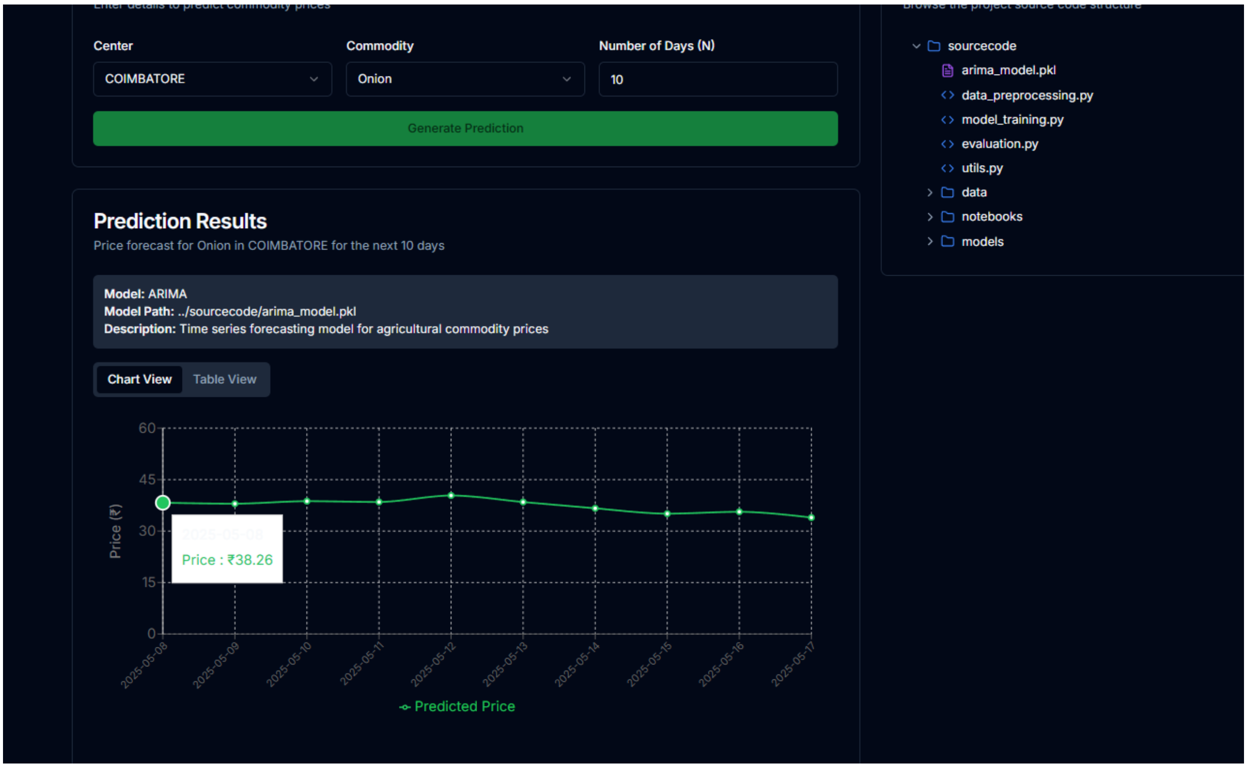


Fig:3 Price Forecasting

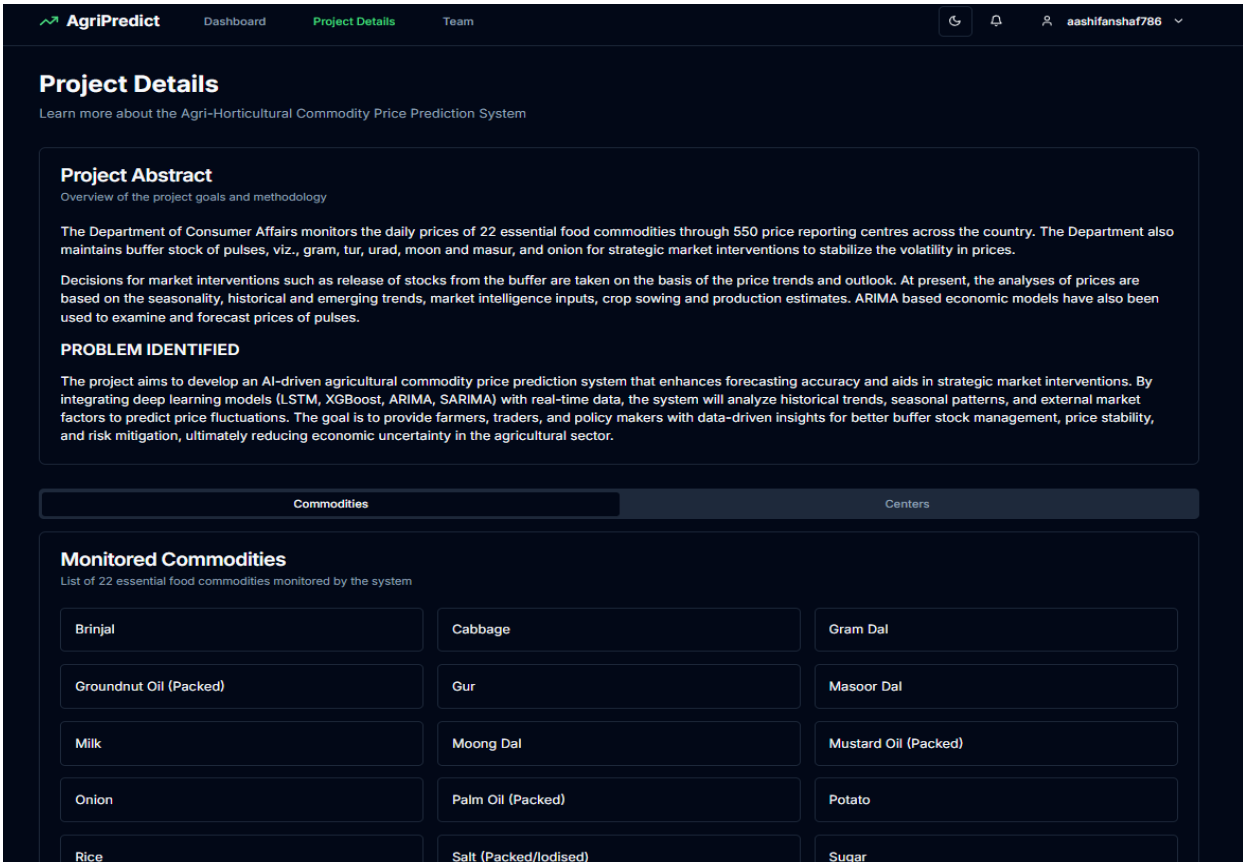
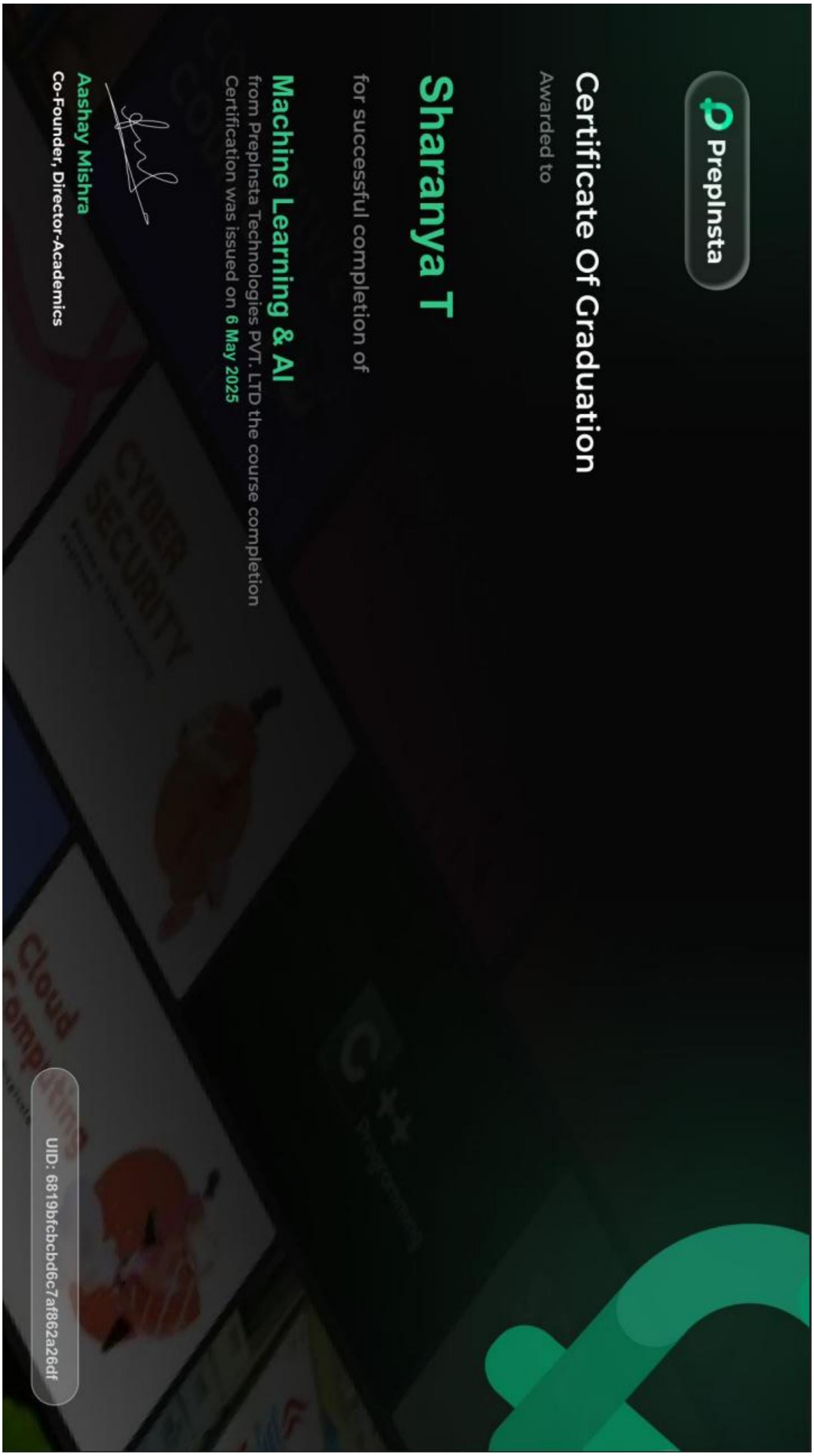
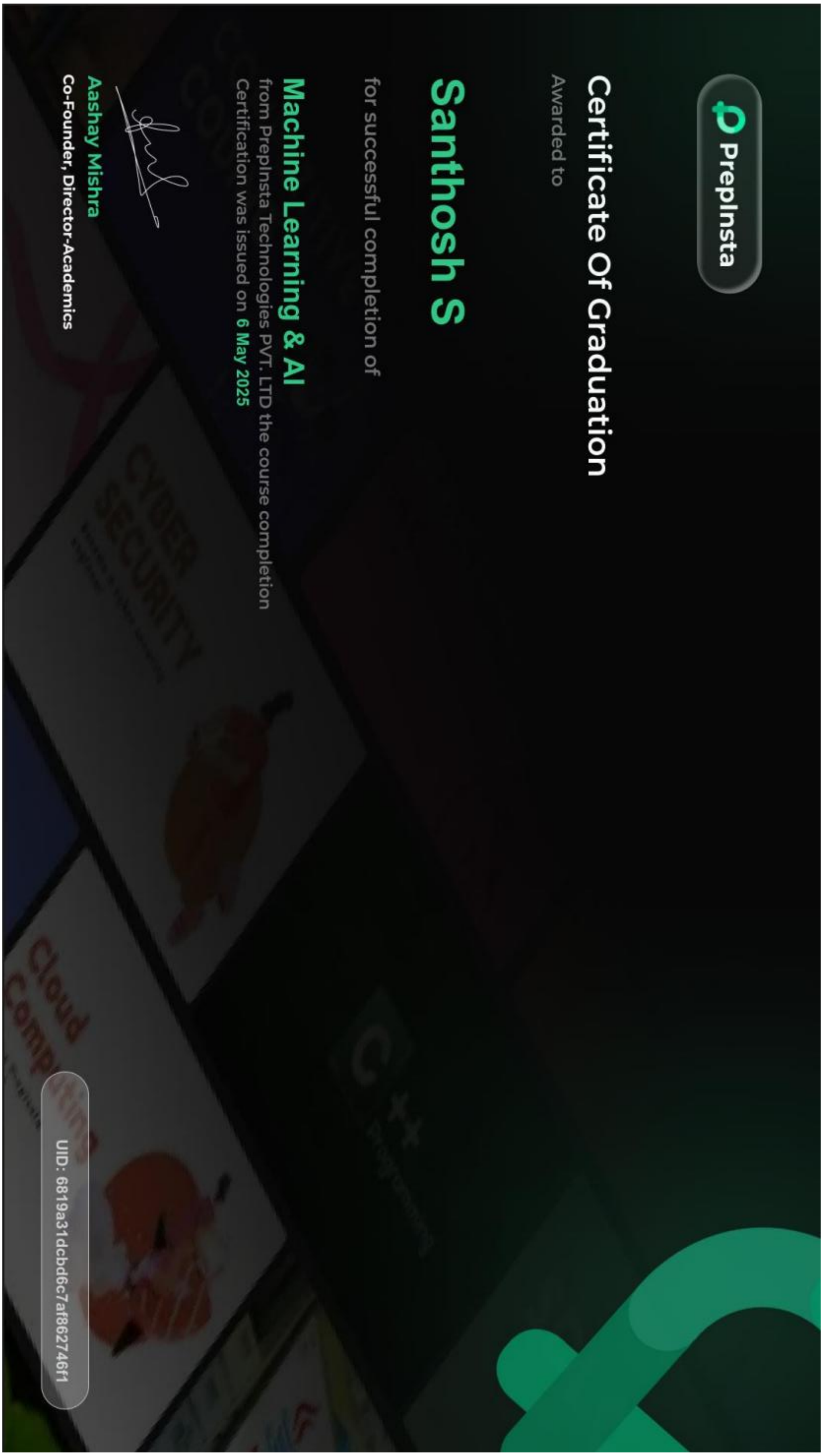
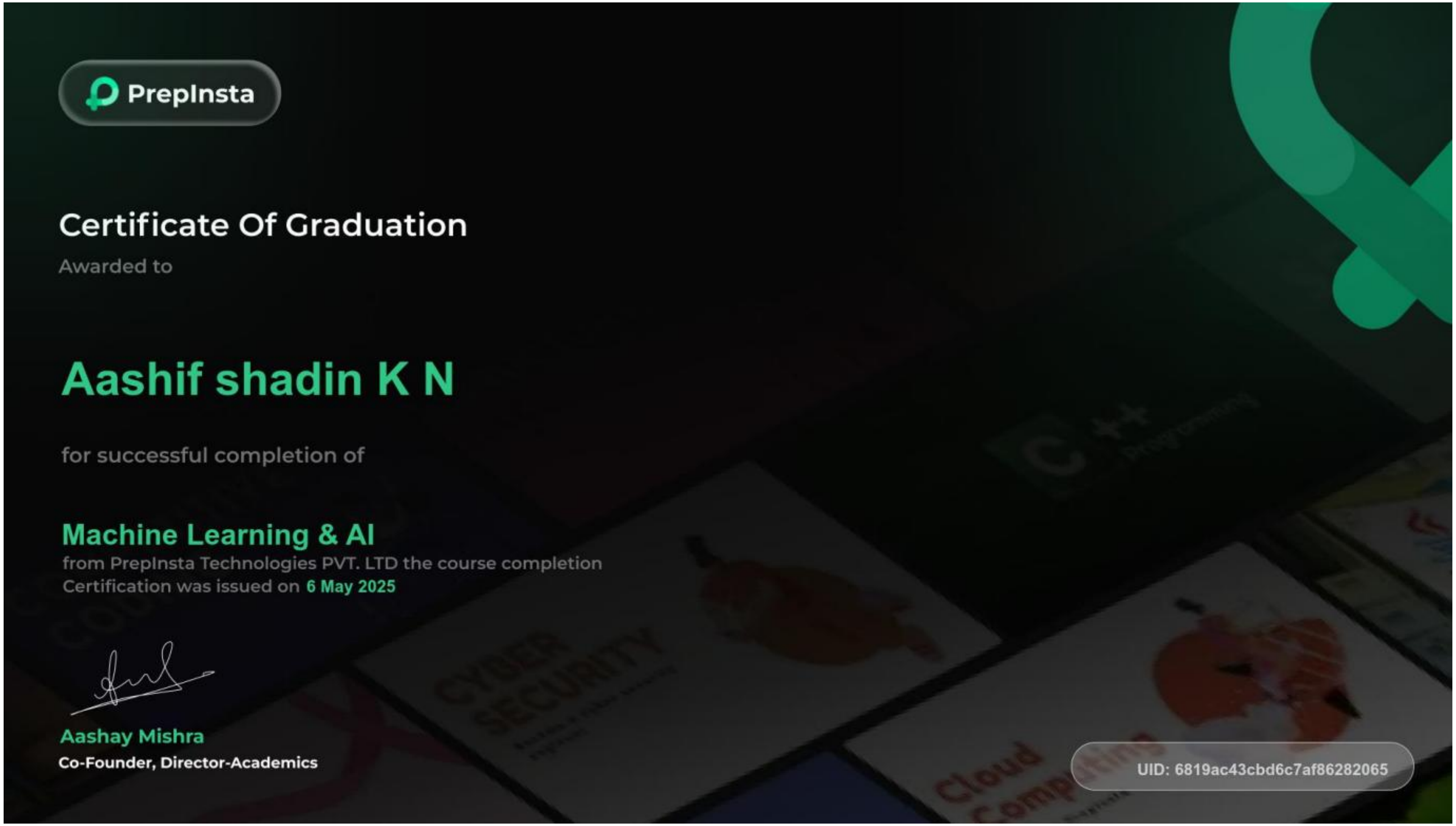


Fig: 4 Project Details

COURSE COMPLETION CERTIFICATES


PrepInsta
Certificate Of Graduation
Awarded to
Sharanya T
for successful completion of
Machine Learning & AI
from PrepInsta Technologies PVT. LTD the course completion
Certification was issued on 6 May 2025
Aashay Mishra
Co-Founder, Director-Academics
UID: 6819bfcbcbd6c7af862a26df


PrepInsta
Certificate Of Graduation
Awarded to
Santhosh S
for successful completion of
Machine Learning & AI
from PrepInsta Technologies PVT. LTD the course completion
Certification was issued on 6 May 2025
Aashay Mishra
Co-Founder, Director-Academics
UID: 6819a31dcbd6c7af862746f1


PrepInsta

Certificate Of Graduation

Awarded to

Aashif shadin K N

for successful completion of

Machine Learning & AI

from PrepInsta Technologies PVT. LTD the course completion Certification was issued on **6 May 2025**

Aashay Mishra

Co-Founder, Director-Academics

UID: 6819ac43cbd6c7af86282065

CONTEST PARTICIPATION

PLAGIARISM REPORT