Tribhuvan University

Faculty of Humanities and Social Sciences

"RENEWABLE ENERGY PRODUCTION PREDICTION"

A PROJECT REPORT

Submitted to

Department of Computer Application

New Summit College

Shantinagar, Kathmandu, Nepal

In the partial fulfillment of the requirements for the Bachelors in Computer Application

Submitted by

Roshan Gurung (T.U. Registration No:6-2-476-54-2019)

October, 2024

Under the Supervision of

Rishav Acharya

Tribhuvan University

Faculty of Humanities and Social Sciences

New Summit College

Supervisor's Recommendation

I hereby recommend that this project prepared under my supervision by **ROSHAN GURUNG** entitled "**RENEWABLE ENERGY PRODUCTION PREDICTION SYSTEM**" in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

SIGNATURE

Rishav Acharya

SUPERVISOR

Department of Computer Application

New Summit College

Shantinagar, Kathmandu, Nepal

Tribhuvan University

Faculty of Humanities and Social Sciences

New Summit College

LETTER OF APPROVAL

This is to certify that this project prepared by ROSHAN GURUNG entitled "**RENEWABLE ENERGY PRODUCTION PREDICTION SYSTEM**" in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

--------------------------- **SIGNATURE** Rishav Acharya Supervisor Department of Computer Application Shantinagar, Kathmandu, Nepal	------------------------- **SIGNATURE** Chok Raj Dawadi Principal New Summit College Shantinagar, Kathmandu, Nepal
------------------------- **SIGNATURE** Internal Examiner	-------------------------- **SIGNATURE** External Examiner

ABSTRACT

The renewable energy production prediction project uses machine learning to forecast a country's future renewable energy output. By incorporating historical data, this system strives to provide accurate predictions and recommendations for energy production levels. The system aims for high forecast accuracy by using Ensemble Learning algorithms, particularly linear regression. It was created as a web-based application using css, react js, and Python, with the primary goal of encouraging investments in renewable energy through reliable predictions. With an expected accuracy of more than 80%, this project has significant potential for guiding both government policies and industrial investments toward a greener and more renewable future.

Keywords: *renewable energy production prediction, linear regression, react js, python, flask etc.*

ACKNOWLEDGEMENT

It gives us enormous pleasure to submit the project report on **" RENEWABLE ENERGY PRODUCTION PREDICTION SYSTEM".** Our report wouldn't have been possible without contribution of few people. At first, we desire to express our deepest sense of gratitude to **Tribhuvan University** for giving us the opportunity to present ourselves this report within the scheduled time. We could not have completed this project without the help from our institution, New Summit College, which supported us academically and provided us with a family-like environment. Then, we would like to express our gratitude & respect to our honorable Principal **Mr. Chok Raj Dawadi,** Co-ordinator **Mr. Prawesh Dhungana** and project Supervisor **Rishav Acharya** for his constant guidance, advice, encouragement & every possible help in the overall preparation of this report. We would also like to express our excessive thanks to our team mate for the excellent support in completing our project. Finally, we are very much grateful to our families who always give us constant support and encouragement. We would like to thank our seniors who helped us greatly to complete this paper. In addition, we will mention our friends who also inspired and helped us to finish our work. We hope that this project paper has been prepared for the fulfillment of the course requirement.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

API	Application programming interface
CSS	Cascading Style Sheet
CASE	Computer-Aided Software Engineering tool
DFD	Data Flow Diagrams
ER	Entity-relationship diagrams
HTTP	Hypertext Transfer Protocol
MASE	Mean Absolute Error
SDLC	Software Development Lifecycle
TWH	Terawatt Hour

CHAPTER 1
INTRODUCTION

1.1 Introduction

The renewable energy production forecasting system leverages historical data and a linear regression model to predict energy output from 2025 to 2050. By utilizing data from a SQLite database, it provides forecasts across various regions and energy sources. A key feature is the inclusion of the Mean Absolute Scaled Error (MASE) metric, which ensures accurate and reliable evaluations of the predictions. This innovative system aims to support proactive planning in the renewable energy sector.

In a world increasingly focused on renewable energy sources, the ability to forecast and anticipate future trends in renewable energy production stands as a critical endeavor. The system under consideration utilizes a proactive strategy to forecast energy production between 2025 to 2050. It does this by utilizing historical data and a basic linear regression model to predict future levels of output from a variety of renewable energy sources.

The renewable energy production forecasting system represents an innovative endeavor aimed at predicting energy generation from 2025 to 2050. Utilizing historical data sourced from a SQLite database, the system employs a linear regression model to extrapolate future renewable energy production across various regions and energy sources. An integral addition to this predictive framework is the incorporation of the Mean Absolute Scaled Error (MASE) computation. The MASE metric serves as a critical tool, enabling a comprehensive evaluation of prediction accuracy and reliability against actual energy production levels.

1.2 Problem Statement

Energy production has always been the most difficult to transition from in this movement. Thus, as a solution, we are carrying out this project to set realistic goals for the future of renewable energy production. The system is able to predict the achievable production of alternative energy up to 2050.

Accurately projecting the future of renewable energy generation until 2050 is a problem in the context of worldwide efforts to decrease carbon footprints. Our goal is to create a system that sets realistic targets for the generation of renewable energy, notwithstanding the complexity and variety of factors impacting energy projects. The objective of this system is to build a prediction model that foretells future trends by analyzing thirty years' worth of historical data on energy output. However, because the factors influencing energy efforts are complex, it is impossible to anticipate with 100% precision. The main goal is to improve forecast accuracy in order to present a thorough picture while taking into account the various strategies that each nation uses to shift to renewable energy sources. Our project mainly creates a pattern of energy production from the past 30 years and then predicts where the pattern is headed towards. This allows us to see a better picture considering each country can have a different approach in this transition. Moreover, this project's success holds the potential to profoundly impact the world. By accurately projecting renewable energy production trends, it can guide policymakers, industry leaders, and environmental advocates in making informed decisions. This knowledge could inform the allocation of resources, support the development of renewable energy policies, and foster collaborations among nations to collectively combat climate change. Ultimately, the implementation of more precise and data-driven strategies in renewable energy production could significantly contribute to mitigating environmental degradation and securing a more renewable future for generations to come.

1.3 Objectives

The main objective of this project are:

- To provide reliable, data-driven insights for stakeholders, policymakers, and energy planners.
- To provides robust predictions for long-term planning.

1.4 Scope and Limitation

The project tries to implement a system that will be able to take in the name of country and year as an input, and provide the results for energy production that includes Hydro, Wind, Solar and Biogas as an output. The system provides output based on data stored in the database. The data in the database is generated based on the calculations where the Linear Regression algorithm is used. Apart from energy prediction, we have implemented error calculation MASE error and confidence level prediction where we predict the level of confidence in producing the predicted amount of energy. The unit of energy is measured in terms of Terawatt Hour (TWH).

1.5 Development Methodology

Iterative Waterfall Model

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

Waterfall model is used in the development of RENEWABLE ENERGY PRODUCTION PREDICTION. The process begins with a comprehensive requirement-gathering phase, where the specific features and functionalities of the renewable energy prediction system are defined in consultation with stakeholders, such as energy providers and analysts. Once these requirements are finalized, the system design phase focuses on creating the system's architecture and data structure, including the integration of prediction algorithms like Linear Regression. After the design is complete, development proceeds according to the specifications, with coding focused on implementing the prediction models and ensuring accurate energy forecasts. Once development concludes, the system undergoes extensive testing to ensure reliability and accuracy in forecasting renewable energy production. This structured, step-by-step approach, following the Waterfall model, ensures a well-documented and stable energy prediction system, utilizing the latest technologies to forecast energy trends accurately.

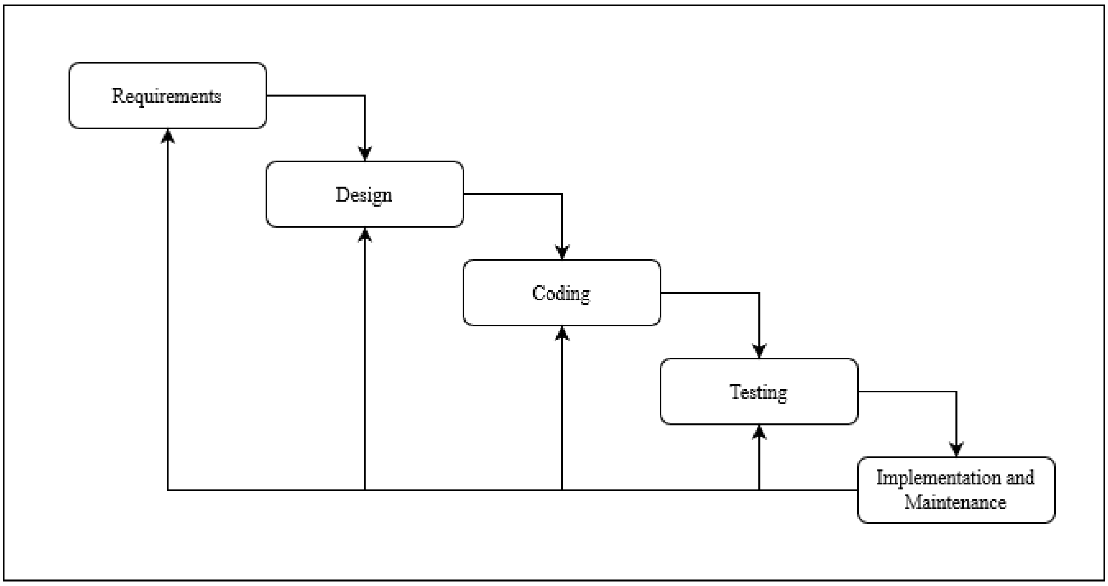


Figure 1.1: Iterative waterfall Model

1.6 Report Organization

The report is organization into 4 chapters

Chapter 1: Introduction:

In this section, the brief introduction of our project, statement of problem and its objectives are discussed.

Chapter 2: Background study and Literature review:

The previous work related to our projects and similar works were studied and different feasibility analysis is summarized in this section.

Chapter 3: System Analysis and Design:

In this section, we have design system architecture, system flow diagram, dataflow diagram etc.

Chapter 4: Implementation and Testing:

In this section, various implementation method and tools are discussed and also contains description of testing.

Chapter 5: Conclusion and Future Recommendation:

This is the final chapter that concludes the project and talks about our future plans with the project.

CHAPTER 2

BACKGROUND STUDY AND LITERATURE REVIEW

2.1 Background Study

In the realm of renewable energy production, predictive modeling plays a crucial role in planning for future energy needs, and reducing environmental impact. Traditional forecasting methods often struggle to capture the complex and dynamic nature of renewable energy sources, such as solar and wind. However, the application of machine learning and data analytics presents a promising approach to improve the accuracy and reliability of these predictions.

A thorough review of the literature reveals a growing trend in the use of machine learning techniques for predicting renewable energy production. These techniques leverage advanced algorithms to analyze historical energy production data, and other relevant variables to forecast future energy output. By incorporating these predictions into energy planning and management systems, stakeholders can make more informed decisions about energy resource allocation, grid management, and infrastructure development.

Previous research in this area has highlighted the importance of data quality and availability, as well as the need for sophisticated modeling techniques to account for the variability and intermittency of renewable energy sources. Additionally, the integration of predictive models with real-time data streams and monitoring systems can further enhance the accuracy and timeliness of energy production forecasts.

Challenges in this field include the need for improved data collection methods, the development of robust modeling frameworks, and the integration of predictive models into existing energy infrastructure. Addressing these challenges will require collaboration between researchers, policymakers, and industry stakeholders to ensure the effective deployment of renewable energy technologies. Looking ahead, continued research and innovation in machine learning and data analytics are expected to drive further advancements in the prediction of renewable energy production. By leveraging these technologies, stakeholders can enhance the efficiency, reliability, and sustainability of energy systems, paving the way for a more renewable future.

2.2 Literature Review

In various evaluation metrics play a pivotal role in assessing the accuracy of predicted outcomes. Metrics such as Mean Absolute Scaled Error (MASE) and confidence levels are crucial in determining the reliability and precision of energy forecasting models. Literature discussing these evaluation methods illustrates their significance in measuring forecast accuracy and estimating the confidence associated with predicted values [1].

The literature surrounding renewable energy production forecasting incorporates diverse methodologies and predictive models employed to anticipate future energy trends. Among these methods, the utilization of linear regression stands out as a prominent technique applied in the realm of energy prediction. Studies exploring the efficacy of linear regression In forecasting renewable energy production emphasize its capability to handle temporal data effectively and provide reasonable predictions. With this in mind, we have utilized the same algorithm for accurate outcome [2].

Providing accessible and user-friendly interfaces for stakeholders to interact with predictive models. These APIs facilitate seamless data retrieval and enhance user engagement by offering an interface to access forecasted energy production data. This API is later used in ReactJS frontend [3].

Data preprocessing techniques, especially those addressing missing values within renewable energy datasets, are pivotal in enhancing the robustness and accuracy of predictive models. Studies in this area emphasize the importance of comprehensive data preparation to ensure the reliability of predictions [4].

CHAPTER 3

SYSTEM ANALYSIS AND DESIGN

3.1 System Analysis

Traditional energy forecasting systems often struggle with accuracy, are time-consuming, and are difficult to access. Many energy analysts rely on outdated methods to predict energy production, which can lead to inefficiencies and unreliable forecasts. The renewable energy prediction system, however, is designed to be user-friendly, efficient, and highly accurate. This system utilizes advanced algorithms, such as Linear Regression, to forecast future energy production, particularly from renewable sources like hydro and solar power. As a web-based application, it allows energy providers, policymakers, and analysts to input historical data and receive detailed predictions on future energy production, enabling better decision-making and resource management in the renewable energy sector.

3.2 Requirement Analysis

Requirement analysis, also known as requirements engineering, is the process of determining user expectations for a new or modified product. It involves frequent communication with the stakeholders and end-users of the product. It involves frequent communication with the stakeholders and end-users of the product to define expectations, resolve conflicts, and document all the key requirements. Requirement analysis allows software engineers to define user needs early in the development process. It helps them deliver a system that meets customer’s time, budget and quality expectations.

3.1.1 Functional Requirements

Functional requirements are product features or functions that developers must implement to enable users to accomplish their tasks. It is a description of the service that the software must offer. It describes the software system or its components. Generally, functional requirements describe a system, its behavior under specific conditions. A function is nothing but inputs to the software system, its behavior, and outputs. It can be calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform.

Functional requirements for the renewable energy production forecasting system include:

1. Data Handling:

- Retrieve historical energy data from a SQLite database.
- Accept country and year as input and provide energy production forecasts (Hydro, Wind, Solar, Biogas).

2. Prediction Modeling:

- Use a linear regression model to predict energy production from 2025 to 2050.
- Ensure predictions include confidence levels and error calculations (MASE).

3. Output Presentation:

- Display energy production results in Terawatt Hour (TWh).

4. Evaluation Metrics:

- Compute and present Mean Absolute Scaled Error (MASE) for prediction accuracy.

5. System Query:

- Allow querying predictions from the database based on user input.

3.2.1 Non-Functional Requirements

1. Accuracy:

- Ensure prediction accuracy using validated error metrics (e.g., MASE).

2. Scalability:

- Support future integration with additional data sources and predictive algorithms.

3. Reliability:

- Ensure system stability and robustness for accurate forecasts over long-term use.

4. Usability:

- Provide a user-friendly interface for input and result interpretation.

5. Performance:

- Process and deliver forecasts promptly.

6. Maintainability:

- Follow a structured design approach (Iterative Waterfall Model) for easy updates and enhancements.

7. Data Integrity:

- Maintain high-quality, validated historical data for accurate predictions.

8. Security:

- Protect the database from unauthorized access or manipulation.

Use Case Diagram

Use Case The following is the use case diagram that describes different functionalities of the system and interaction between actors:

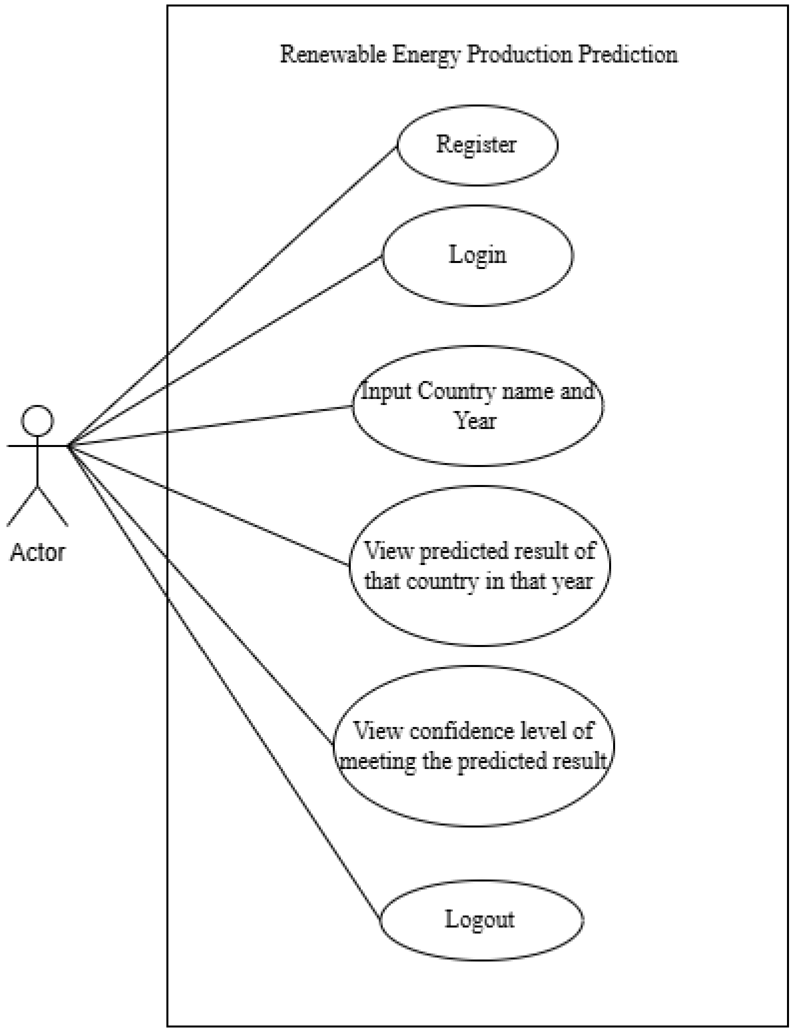


Figure 3.1:Use Case Diagram

Use Case Description:

Table 1: Login

Use case identifier	**UC1 - Login**
Primary Actor	User
Secondary Actor	None
Description	User logs into the system.
Pre-condition	User isn't logged in.
Success-scenario	User successfully logs in.
Failure-scenario	User fails to log in.

Table 2: Register

Use-Case Identifier	**UC2 – Register**
Primary Actor	User
Secondary Actor	None
Description	User registers in the system.
Pre-condition	User is not registered in the system.
Success-scenario	User successfully registered
Failure scenario	User fails to register in the system.

Table 3: User Inputs

Use-Case Identifier	**UC3 – Input Customer Detail**
Primary Actor	User
Secondary Actor	None
Description	User enters the name of country and year.
Pre-condition	User has not entered their details in the system.

Success scenario	User successfully enters the data in the input box.
Failure scenario	User fails to enter the details in the system.

Table 4: View the predicted energy production

Use-Case Identifier	UC4 – View the predicted energy production
Primary Actor	CUSTOMER
Secondary Actor	None
Description	User views the predicted renewableenergy production for that particular year.
Pre-condition	User has not viewed the predicted renewable energy production for that particular year.
Success scenario	User views the predicted production ofthe renewable energy for that year successfully.
Failure scenario	User fails to view the predicted production of renewable energy.

Table 5: Analyze the predicted data

Use-Case Identifier	UC5 Analyze the predicted data
Primary Actor	User
Secondary Actor	None
Description	User views the probability of achieving the predicted renewable energy production for thatparticular year.

Pre-condition	User has not viewed the probability of achieving the predicted renewable energy productionfor that particular year..
Success scenario	User views the probability of achieving the predicted production of the renewable energy for that year successfully.
Failure scenario	User fails to view the probability of achieving the predicted production of renewable energy.

3.3 Feasibility Study

Feasibility study for the renewable energy production forecasting project involves evaluating its technical, economic, and operational viability:

Technical Feasibility:

- **Data Availability:** Availability of historical energy production data up to 2020 in a suitable format (.csv) for analysis.
- **Algorithm Suitability:** Suitability of linear regression algorithms for accurately predicting renewable energy production.
- **System Compatibility:** Ensure the system is compatible with required programming languages (Python), libraries (Pandas, NumPy), and frameworks (Flask).

Economic Feasibility:

- **Resource Allocation:** Assessment of resources required for system development, including infrastructure (computing resources, storage), personnel, and software tools.
- **Cost-Benefit Analysis:** Evaluate the cost-effectiveness of implementing the system against the potential benefits derived from accurate energy production predictions.
- **ROI Estimation**: Estimate the return on investment considering the value derived from improved energy planning and decision-making.

Operational Feasibility:

- **User Acceptance:** Analyze user requirements and expectations to ensure the system aligns with user needs.

- **Ease of Use:** Assess the system's user interface and interaction flow for usability and user-friendliness.
- **Scalability:** Ensure the system is scalable to accommodate increasing data volume and user demands.

Gantt Chart

Table 6: Gantt Chart

Working Time	10th Jul	1th Jul	5th Aug	15th Aug	29th Aug	20th Sep
Requirement						
Design						
Coding						
Testing						
Implementation						
Documentation						

3.4 Object Modeling

Class Diagram

The main purpose of Diagram for our project is to build a static view of an application. It is a blueprint of a system as it visualizes the particular working functionality of our system.

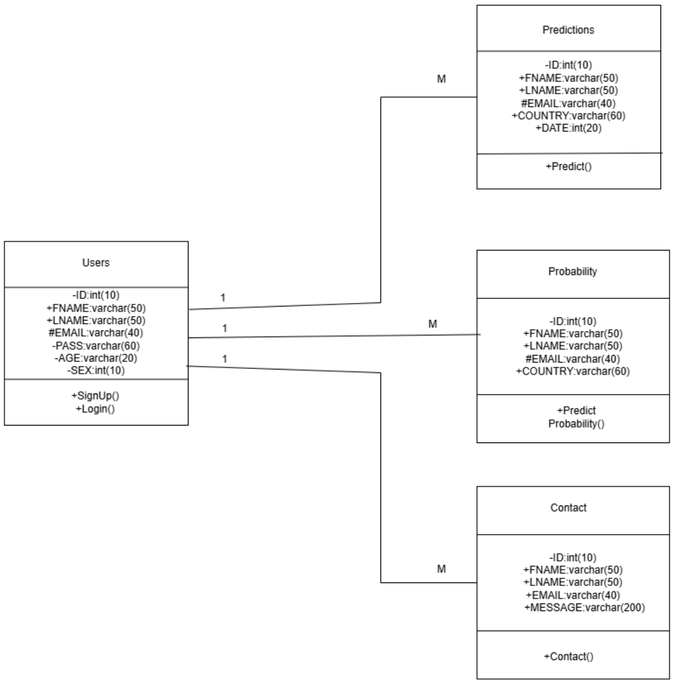


Figure 3. 2:Class Diagram

3.5 Dynamic Modeling

A dynamic model represents the time-dependent aspects of a system. In our system sequence diagram simply depicts interaction between objects in a sequential search.

Sequence diagram

The Sequence Diagram outlines how users interact with the renewable energy prediction system. It illustrates key actions such as logging into the system, viewing historical energy data, selecting energy sources (like hydro or solar), and generating predictions for future energy production. This sequence helps users efficiently navigate the system to obtain accurate energy forecasts.

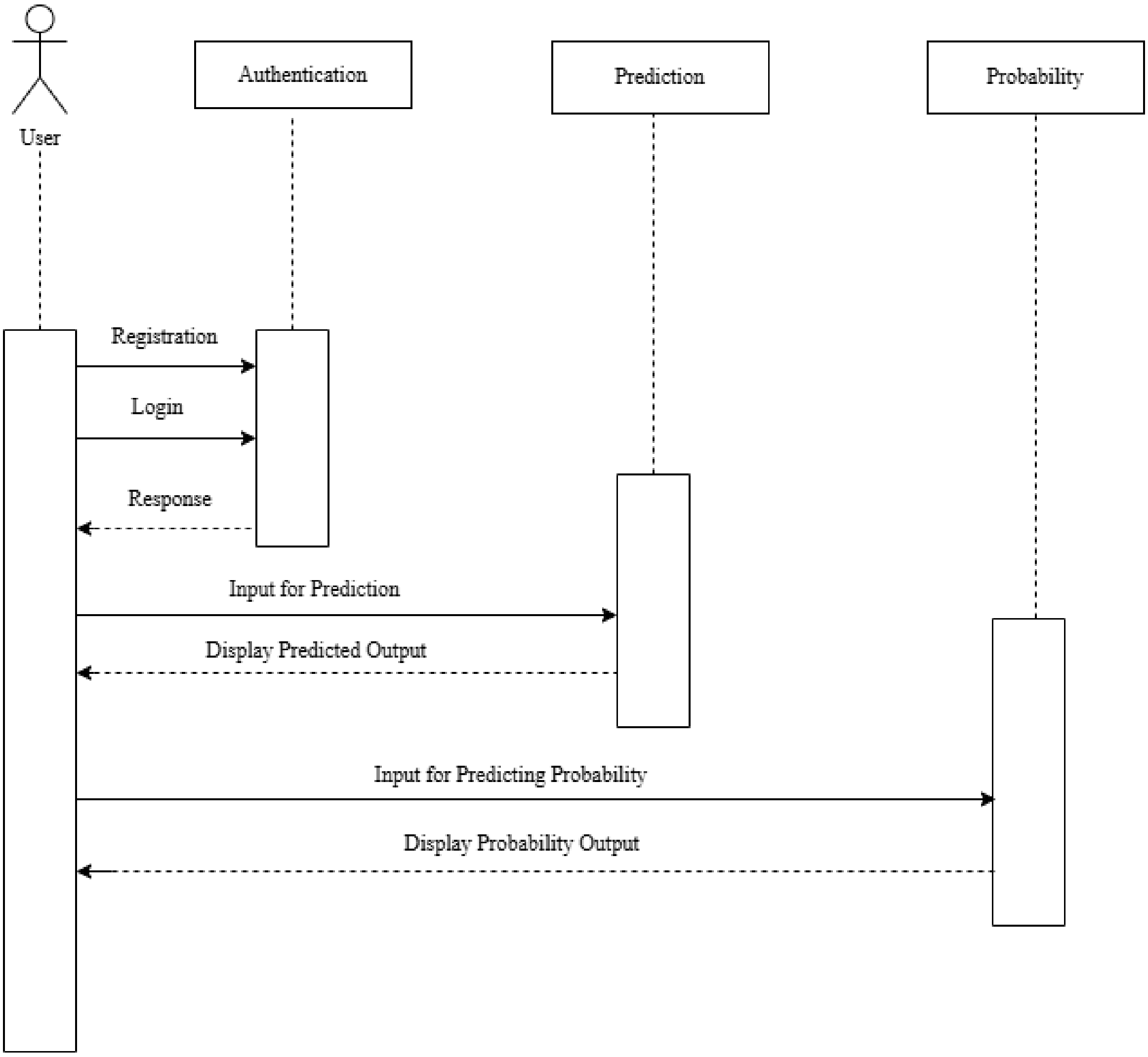


Figure 3. 3:Sequence Diagram

3.6 Process Modelling

Activity Diagram

The activity diagram shows the diagram view of the system. Moreover, it is also represented as the flowchart which represents the flow of control among the activities of the system. In Prediction System, the activity diagram shows the user in the diagrammatic form.

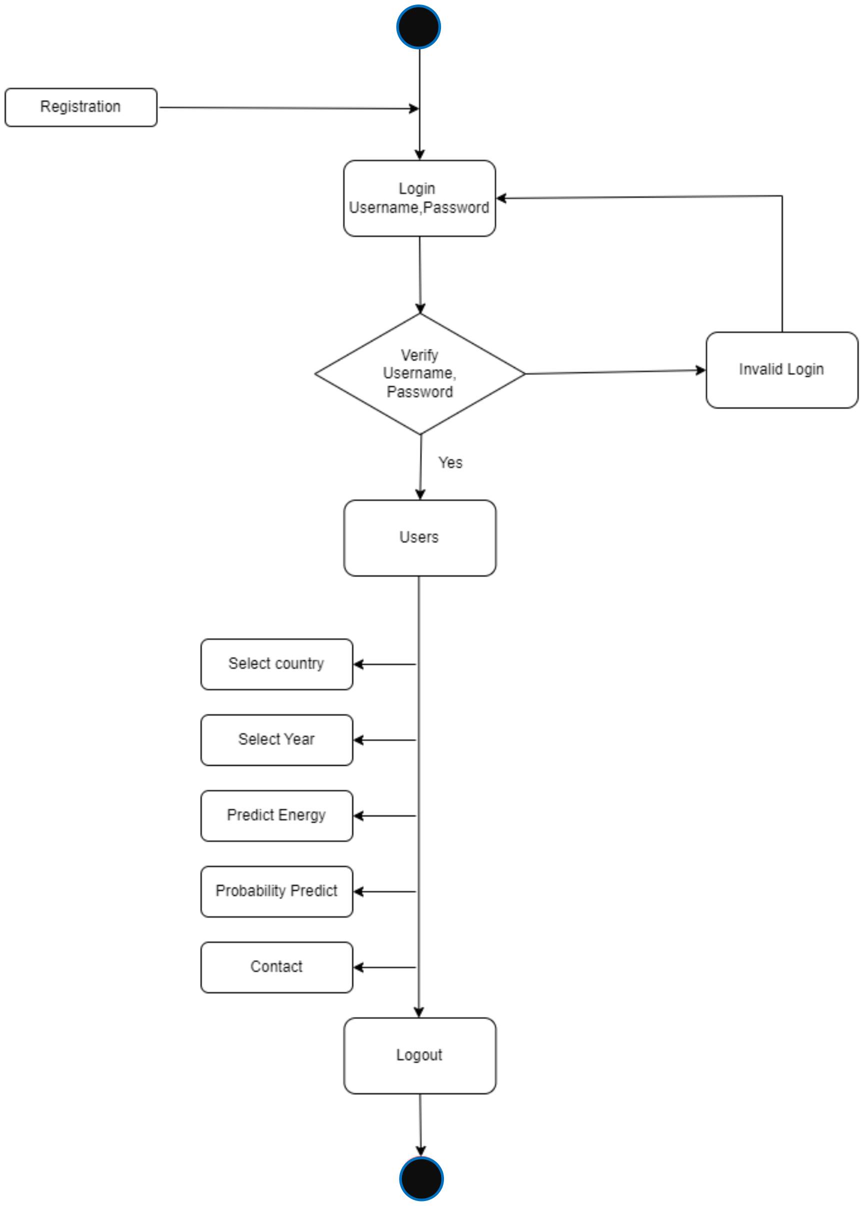


Figure 3. 4:Activity Diagram

3.7 System Design

It is the process of defining the element of a system such as the architecture, modules and the components the different interface of those components and the data that goes through the system.

Component Diagram

The Component diagram helps to visualize the physical components of the system and their dependency relationship. The RENEWABLE ENERGY PRODUCTION PREDICTION has the components such user parties involved to make a complete architecture.

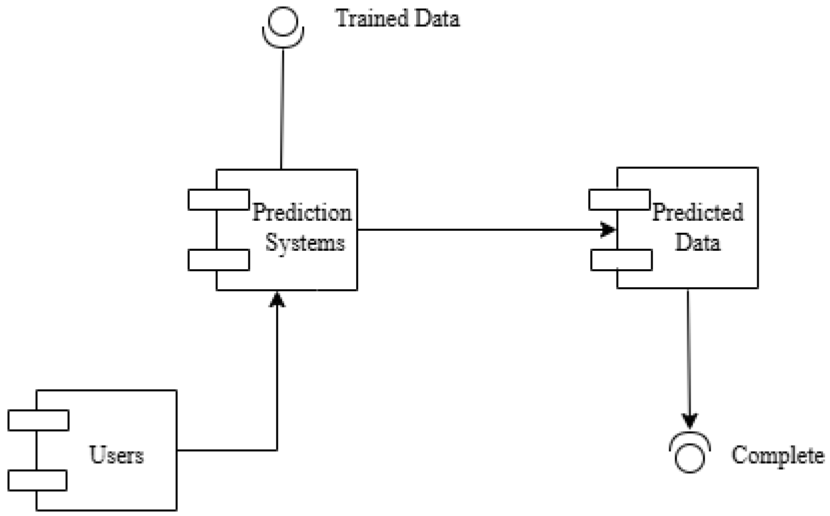


Figure 3. 5:Component Diagram

Deployment diagram

Deployment diagram helps to visual representation of the physical deployment of software component and hardware nodes in a system. This diagram Clarify the how various system components are distributed and connected.

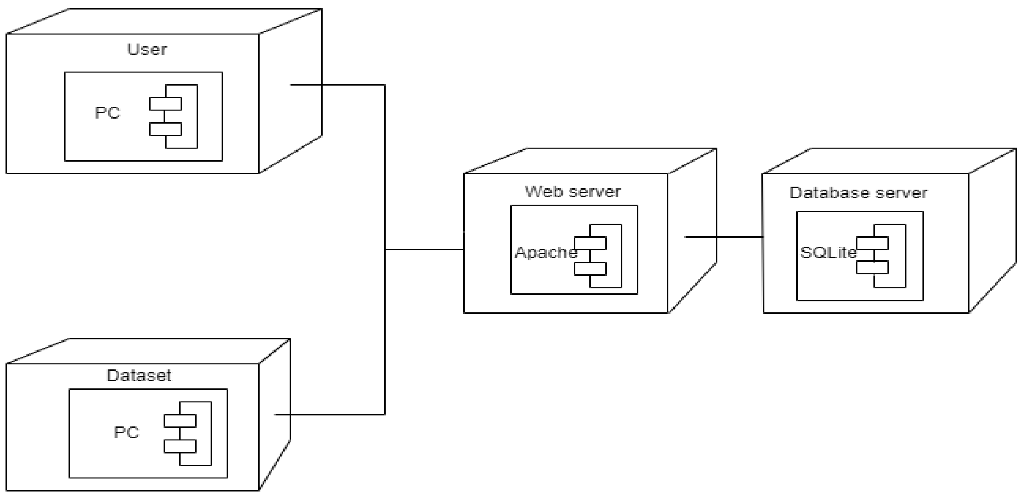


Figure 3. 6:Deployment Diagram

3.8 Algorithm Details

Linear Regression is a simple and widely-used machine learning algorithm that models the relationship between a dependent variable (the variable to be predicted) and one or more independent variables (the input features). It is a type of supervised learning algorithm, commonly used for regression tasks, where the goal is to predict continuous values.

The linear regression algorithm finds a straight line that best fits the given data points. The line is defined by the equation:

$$y = b0 + b1.x$$

Where
y: Predicted value (dependent variable).
X: Independent variable.
$b0$: Intercept (value of y when X=0).
$b1$: Slope (rate of change of y with respect to X).

Step 1: Input

A dataset with X (Independent variable) and Y (dependent variable).

Step 2: Campute Means

- Compute the mean of the X values (X mean) and Y values (Y mean).
- These means represent the average of the independent and dependent variables.

Step 3: Compute the Slope ($b1$)

$$b_1 = \frac{\sum(X_1 - X_{mean}) \cdot (Y_i - Ymean)}{\sum(X_i - X_{mean})^2}$$

- Measure how Y changes for every unit increases in X (numerator).
- Divides this by the spread of X values (denominator).

Step 4: Compute the intercept(b0)

The intercept b0 is calculated using:

$$b_0 = y_{mean} - b_1 \cdot x_{mean}$$

Step 5: Generate the regression line

The equation $y = bo + b_1 \cdot X$ to predict y for any given X.

Step 6: Output

Example

Input Data:

X (Year)	Y (Electricity)
2000	25
2005	30
2010	35
2015	40
2020	50

Calculate Means

$$x_{\text{mean}} = \frac{2000+2005+2010+2015+2020}{5} = 2010$$

$$y_{\text{mean}} = \frac{25+30+35+40+50}{5} = 36$$

Calculate Slope(b1)

$$\text{Numerator} = \sum (X_1 - X_{mean}) \cdot (Y_i - Ymean)$$

=(2000−2010)(25−36)+(2005−2010)(30−36)+(35-36)+(40-36)+(2020−2010)(50−36)
=(−10)(−11)+(−5)(−6)+(0)(−1)+(5)(4)+(10)(14)=110+30+0+20+140=300

$$\text{Denominator} = \sum (X_i - X_{mean})^2$$

$$= (2000 - 2010)^2 + (2005 - 2010)^2 + (2010 - 2010)^2 + (2015 - 2010)^2 (2020 - 2010)^2$$

$= (-10)^2 + (-5)^2 + (0)^2 + (5)^2 + (10)^2 = 100 + 25 + 0 + 25 + 100 = 250$

$$b1 = \frac{\text{Numerator}}{\text{Denominator}} = \frac{300}{250} = 1.2$$

Calculate Intercept(b0)

$$b_0 = y_{\text{mean}} - b_1 \cdot \text{Xmean}$$

$$b_0 = 36 - (1 \cdot 2.2010) = 36 - 2412 = -2376$$

Regression Line

$$y = \ -2376 + 1.2. X$$

Prediction

Predict electricity of 2025, or X=2025

y = -2376+1.2. 2025 = -2376+2430 = 54 TWH

Applying the Algorithm

Steps:

1. Start
2. Data Retrieval and Preprocessing:
   - ➔ Load historical data from 'renewable.csv'.
   - ➔ Drop missing values.
3. Linear Regression and Prediction:
   - ➔ Group data by 'Entity' (Region/Country).
   - ➔ For each entity:
     - ➔ Perform linear regression for each energy source (hydro, wind, solar,etc.) separately.
     - ➔ Predict energy production for the years 2025 to 2050.
     - ➔ Adjust predictions to ensure non-negativity.
4. MASE Calculation:
   - ➔ Evaluate prediction accuracy using MASE.
   - ➔ Calculate MASE for each region and energy source.
   - ➔ Store MASE values in 'mase__error.db'.
5. Probability Calculation:
   - ➔ Implement the probability calculation program.
   - ➔ For each entity:
     - ➔ Compute prediction intervals based on linear regression predictions.
     - ➔ Calculate probabilities and confidence levels for predicted outcomes.
     - ➔ Store probability values in 'confidence_levl.db'.
6. Database Operations:
   - ➔ Store predictions, probabilities, and confidence levels in respectivedatabases.
7. API Creation with Flask:
   - ➔ Implement Flask APIs to access prediction data, confidence levels, andMASE errors.
   - ➔ /get_predictions - Retrieve predictions based on entity name.

→ /get_confidence_level - Retrieve confidence levels based on entity name.

→ /get_mase_error - Retrieve MASE errors based on entity name.

8.End

Algorithm Pseudo Code

```
def LinearRegressionFit(X, y):
    n = len(X)
    x_mean = sum(X) / n
    y_mean = sum(y) / n

    numerator = sum((X[i] - x_mean) * (y[i] - y_mean) for i in range(n))
    denominator = sum((X[i] - x_mean) ** 2 for i in range(n))

    b1 = numerator / denominator
    b0 = y_mean - b1 * x_mean

    return b0, b1
```

CHAPTER 4

IMPLEMENTATION AND TESTING

4.1 Implementation

In the implementation of the renewable energy production prediction project, we have utilized various tools and technologies, employing them for specific tasks throughout the project. Here's how we employed these tools in the project:

4.1.1 Tools Used

Draw.io:

We utilized Draw.io, an online diagramming tool categorized as a Computer-Aided Software Engineering (CASE) tool. Draw.io played a significant role in creating visual representations such as flowcharts, system architecture diagrams, context diagrams, data flow diagrams (DFD), entity-relationship (ER) diagrams, and other schematic illustrations. We employed Draw.io to design and document various components and processes within the system. This tool enabled us to create detailed diagrams depicting the system architecture, data flow, database schema, user interactions, and other essential aspects of the project. Draw.io' s intuitive interface and diverse set of shapes, symbols, and connectors facilitated the creation of clear and comprehensive visual models.

Through Draw.io, we developed ER diagrams showcasing the relationships between entities in the database, system flowcharts delineating the workflow of data and processes, DFDs illustrating the data movement and transformations, and other visual representations that aided in better comprehension and communication of the project's architecture and functionalities. The diagrams created using Draw.io were crucial in visualizing the system's components, thereby aiding in the design, development, and documentation phases of the project.

Python:

Python was the core programming language used throughout the project. Leveraging libraries such as Pandas, NumPy, Scikit-learn, and Matplotlib, I performed data preprocessing, developed predictive models using linear regression, conducted data visualization, and evaluated model performance.

Jupyter Notebook:

We have extensively utilized Jupyter Notebook as an interactive platform for data exploration, model prototyping, and analysis. This allowed for an iterative approach to 22 model development, enabling step-by-step execution and documentation of code logic and outcomes.

Flask:

For creating APIs to serve predictions and related data, we employed Flask, a micro web framework in Python. Flask facilitated the development of endpoints to handle HTTP requests, providing users or client applications with access to stored predictions, confidence levels, MASE errors, and other relevant data stored in the SQLite3 database. The data is returned to user upon POST request.

SQLite3 Database:

SQLite3 was used as the database management system to store and manage prediction results, MASE errors, confidence levels, and other pertinent data. The lightweight and self-contained nature of SQLite3 allowed for seamless integration with Python, enabling efficient data storage and retrieval.

By utilizing these tools and technologies in the project, we have successfully developed a system that encompasses data preprocessing, model training, storage of prediction results and related metrics, API creation, and database management for serving the predictions and associated data to end-users.

Dependencies

Dependencies are libraries, modules, or packages that a program relies on to function correctly. These dependencies also provide additional functionality or resources that the program needs but does not include directly in its codebase.

- **numpy:** Offers support for numerical computing with powerful array operations and linear algebra capabilities.
- **Flask:** A high-level Python web framework for building web applications quickly with a clean and pragmatic design.

React:

React is a popular JavaScript library used for building user interfaces. It allows developers to create interactive and dynamic UI components for web applications. React uses a component-based architecture, where UI elements are broken down into reusable and independent components. This makes it easier to manage complex UIs and encourages code reusability.

CSS:

CSS, or Cascading Style Sheets, is a style sheet language used to describe the presentation of a document written in HTML or XML. It defines how elements should be displayed on a webpage, including layout, colors, fonts, and more. CSS allows developers to separate the content of a webpage from its presentation, making it easier to maintain and update the design of a website.

JavaScript:

JavaScript is a versatile scripting language commonly used to create interactive and dynamic behavior on web pages, enabling features like form validation, animations, and DOM manipulation to enhance user experience.

Bootstrap:

Bootstrap is a free and open-source web development framework. It’s designed to ease the web development process of responsive, mobile-first websites by providing a collection of syntax for template designs.

In other words, Bootstrap helps web developers build websites faster as they don’t need to worry about basic commands and functions. It consists of HTML, CSS, and JS-based scripts for various web design-related functions and components.

4.2 Testing

In the testing phase of the renewable energy production prediction program, we conducted rigorous evaluations encompassing both unit testing and system testing to ensure the reliability, accuracy, and robustness of the developed system.

4.2.1 Unit testing

Unit testing focuses verification effort on the smallest unit of software design- the software component or module. The unit test is white-box oriented. The unit testing implemented in every module of Energy Prediction System, by giving correct manual input to the system, the data are stored in database and retrieved. If you want required module to access input or get the output from the End user any error will accrue the time will provide handler to show what type of error will be accrued.

Test Case for Unit Testing

Table 7: Unit testing

S.N	Test Case ID	Test Description	Steps Executed	Excepted Result	Actual Result	Pass/ Fail
1.	TC-01	Open the System UI page	Run the Flask server in backend and start the NPM	UI screen should be displayed.	UI screen was displayed.	Pass
2.	TC-02	Open the user Registration page	Click the user registration icon and fill up the page	Registration should be successful.	Registration should be successful.	Pass
3.	TC-03	Open the user login page	Click the user login icon and enter the username and password	Login should be successful and user profile should be accessed.	Login was successful and user profile was displayed.	Pass
4.	TC-04	Open the user login page with wrong username and password.	Click the user login icon and enter the wrong username and password Eg: Username: Ram Password:8888	Login should be successful.	Login was not successful and get error message “invalid username and password	Pass
5.	TC-05	To Predict the energy sources.	Select the country and the year and click the predict button.	Pie chart should be Displayed.	Pie chart was Displayed with percentage.	Pass

6.	TC-06	To send the message to Admin page	Click the contact button and write something and send the message to the admin	Message should be sent to the admin page.	Message was not Send to the admin button.	Pass
7.	TC-07	Logout	Click Logout Button	Should be redirect to UI page.	Was redirect to login page.	Pass

4.2.2 System Testing:

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Below we have described the two types of testing which have been taken for this project. It is to check all modules worked on input basis. if you want change any values or inputs will change all information. So specified input is must.

Test case for System Testing

Table 8: Test Case for System Testing

S.N	Test Case ID	Test case name	Test Description	Steps Executed	Excepted Results	Actual Results	Pass/ Fail
1.	TC-01	Security testing	Checking a security to user login	Login with registered username and password	Login should be successful and the user profile should be accessed	Login was successful and user profile was displayed	Pass

2.	TC-2	Security testing	Checking a security with invalid username and password.	Login with invalid username and password.	Login should be successful and the user profile should be accessed	Login was not successful	Pass
3.	TC-3	Usability testing	Select the country and date	Click the apply button	Output should be displayed in Pie Chart	Output was displayed in Pie Chart.	Pass
4.	TC-4	Message Testing	Sending Message	Type message or contact details and send to the verified email address.	Message should be send to the verified email address.	Message was sent to the verified email address.	Pass

Throughout the testing phase, we documented the test cases, recorded test results, and iteratively refined the program based on identified issues or discrepancies. The objective was to ensure that the system adhered to the specified requirements, functioned reliably under various conditions, and delivered accurate predictions and confidence levels to the end-users.

CHAPTER 5

CONCLUSIONS AND FUTURE RECOMMENDATIONS

5.1 Conclusion

Linear Regression algorithm predicts the production of renewable energy accurately. It allows researchers and engineers to set realistic goals for the near future as we are continuing to take big strides for carbon neutral energy.

The inclusion of the probability-checking program within the predictive system adds an essential layer of evaluation, providing for a more nuanced knowledge of the possibility of anticipated energy production results. While the system correctly anticipates energy production patterns, the probability assessments provide stakeholders with information into the likelihood of these projections being met.

The integration of MASE within the predictive system presents an invaluable enhancement, offering a standardized means to evaluate forecasting accuracy. By combining historical data retrieval, linear regression modeling, and MASE computation, the system has provided insights into future energy production trends while offering a reliable method for assessing predictive performance.

Recognizing and fixing the model's limits and data quality concerns will be critical for improving predicted accuracy in the future. The model's ongoing improvement and future adoption of sophisticated approaches will strengthen its ability to give more accurate projections, allowing for more informed decision-making in the domain of renewable energy planning and sustainability programs.

In conclusion, the integration of MASE within the renewable energy predictive system signifies a significant step towards assessing and enhancing forecasting accuracy. The system's ability to not only predict energy production trends but also evaluate its own predictive performance offers substantial benefits. Despite facing challenges related to model limitations and data quality, the inclusion of MASE contributes to informed decision-making in renewable energy planning and policy formulation. Moving forward, continual refinement and adaptation of predictive models will be pivotal in ensuring more accurate and reliable forecasts, thereby supporting renewable energy initiatives on a global scale.

5.2 Future Recommendations

A number of important factors need to be taken into account while developing a linear regression-based system to anticipate the output of renewable energy. The quality and diversity of the data that are incorporated into the model are of utmost importance. This entails taking into account a broad range of elements, including pertinent variables and historical data on energy output. Enhancing the prediction power of the linear regression model can be achieved by guaranteeing an extensive dataset.

It is essential to use feature engineering approaches to draw meaningful conclusions from the data. In order to do this, features that are most pertinent to the estimation of renewable energy production must be transformed and chosen. The model can better capture the underlying patterns and relationships in the data by carefully choosing and engineering features, which will produce predictions that are more accurate.

References

[1] A.Ahuja, "Mean Absolute Scaled Error(MASE) in Forecasting," 2021.

[2] Deepanshi, "All you need to know about your first Machine Learning model – Linear Regression," *Analytics Vidya,* 2023.

[3] J. Dholakia, "Creating RESTful Web APIs using Flask and Python," 26 04 2020. [Online]. Available: https://towardsdatascience.com/creating-restful-apis-using-flask-and-python-655bad51b24. [Accessed 2024].

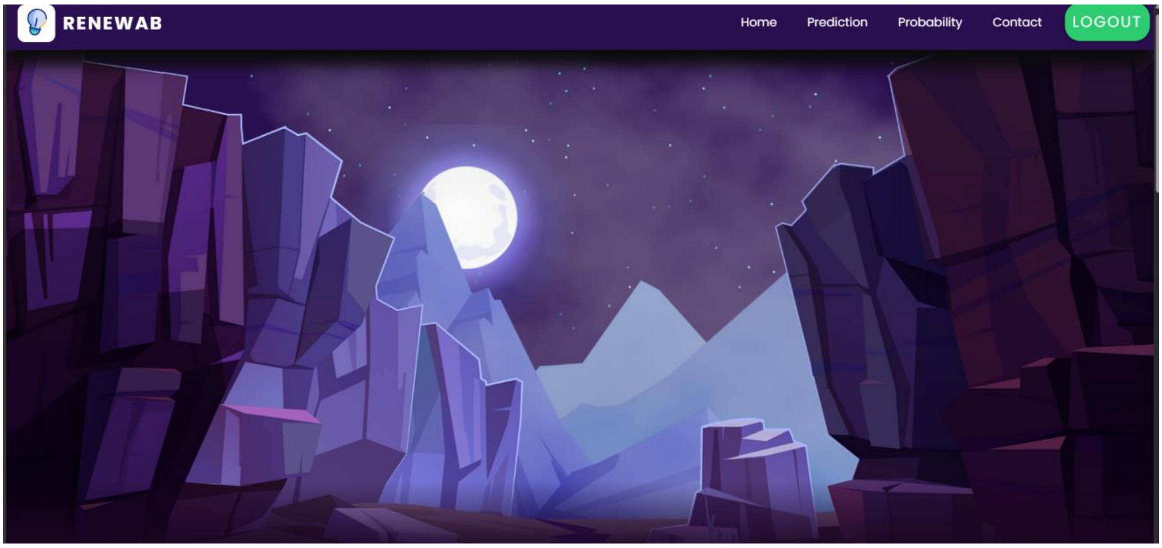
[4] S. Learn, "What Is Data Processing," 13 Jan 2024. [Online]. Available: https://www.simplilearn.com/what-is-data-processing-article. [Accessed 2024].

[5] IBM, "What is linear regression?," [Online]. Available: https://www.ibm.com/topics/linear-regression. [Accessed 03 Jan 2024].

[6] Stackoverflow, "How to correctly use react-router," Empowering the world to develop technology, [Online]. Available: https://stackoverflow.com/questions/37521557/how-to-correctly-use-react-router. [Accessed 3 Jul 2024].

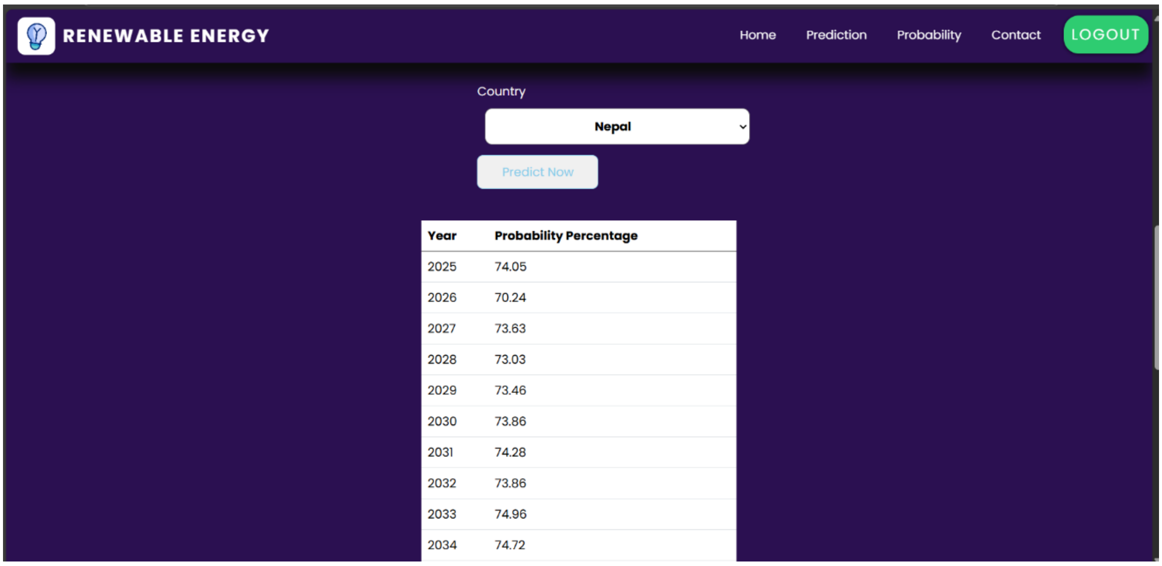
APPENDICES

APPENDICES A: Screenshots

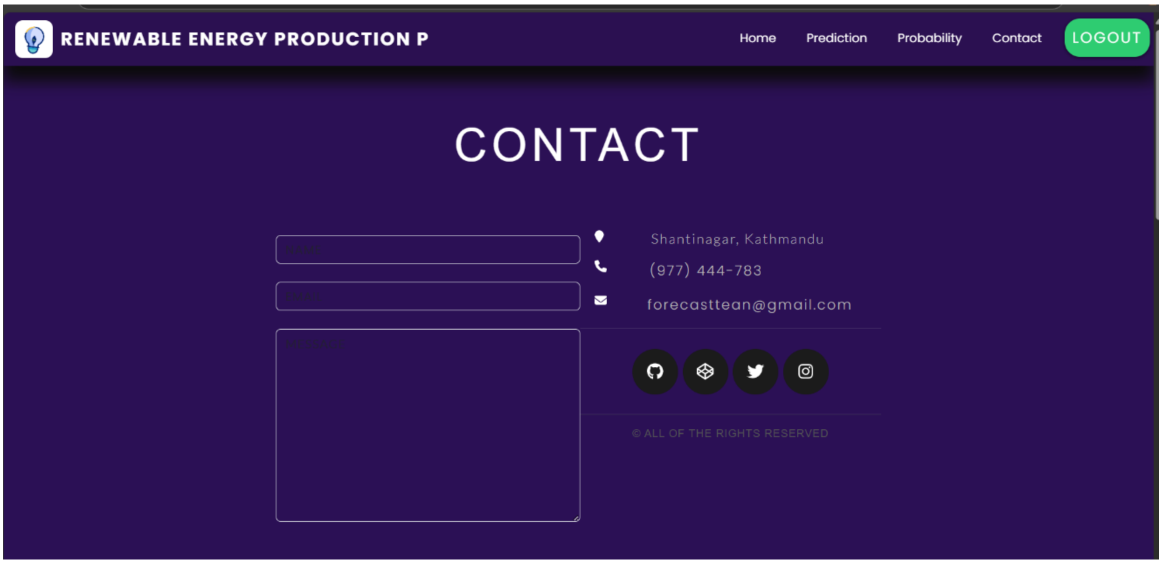


Appendix 1: This is Home page of the system

Appendix 2: This is the Prediction page of the system



Appendix 3: This is the Probability testing page of the system



Appendix 4: This is the Contact page of the system

APPENDICES B: Source Code

```
from flask import Flask, request, jsonify
import sqlite3
import pandas as pd
import json
from flask_cors import CORS

app = Flask(__name__)
CORS(app, resources={r"/*": {"origins": "*"}})


@app.route('/get_predictions', methods=['POST'])
def get_predictions():
    try:
        entity_name = json.loads(request.data.decode("UTF-8"))["entity_name"]

        print(entity_name)

        conn = sqlite3.connect('renewable_predictions_test.db')

        query = f"SELECT * FROM predictions WHERE Entity = ?"
        result = pd.read_sql_query(query, conn, params=(entity_name,))


        conn.close()


        if not result.empty:
            return jsonify(result.to_dict(orient='records'), result.to_dict(orient='records'))
        else:
            return jsonify({'message': 'Entity not found'}), 404

    except Exception as e:
        return jsonify({'error': str(e)}), 500
    pass

@app.route('/get_confidence_level', methods=['POST'])
def get_confidence_level():
    try:
        entity_name = json.loads(request.data.decode("UTF-8"))["entity_name"]

        print( entity_name)

        conn = sqlite3.connect('confidence_levels.db')
```

```
        query = "SELECT * FROM Predictions WHERE Entity = ?"
        result = pd.read_sql_query(query, conn, params=(entity_name,))

        conn.close()
        if not result.empty:
            return jsonify(result.to_dict(orient='records'))
        else:
            return jsonify({'message': 'Entity not found'}), 404

    except Exception as e:
        return jsonify({'error': str(e)}), 500


@app.route('/get_mase_error', methods=['POST'])
def get_mase_error():
    try:
        entity_name = json.loads(request.data.decode("UTF-8"))["entity_name"]

        print( entity_name)

        conn = sqlite3.connect('mase__error.db')
        query = "SELECT * FROM MASE_ERROR WHERE Entity = ?"
        result = pd.read_sql_query(query, conn, params=(entity_name,))

        conn.close()

        if not result.empty:
            return jsonify(result.to_dict(orient='records'))
        else:
            return jsonify({'message': 'Entity not found'}), 404

    except Exception as e:
        return jsonify({'error': str(e)}), 500


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