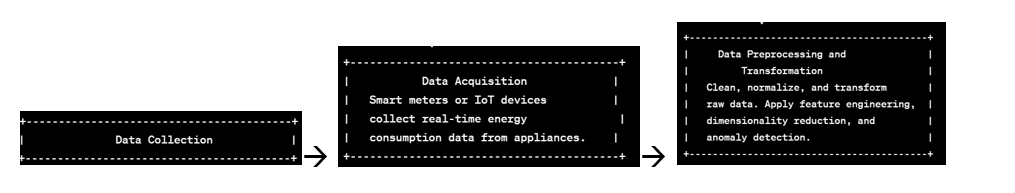
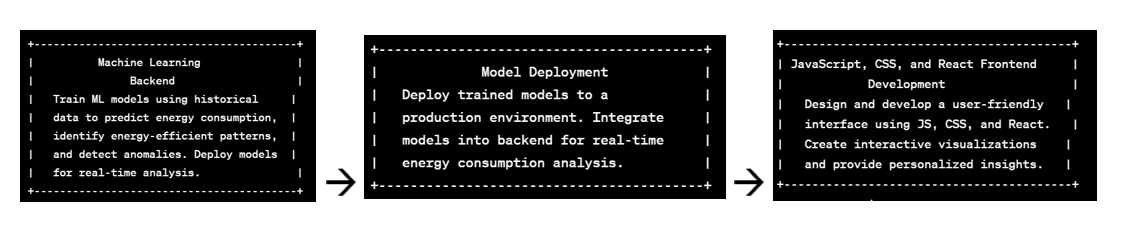
**A Reliable Energy Consumption Analysis System for**

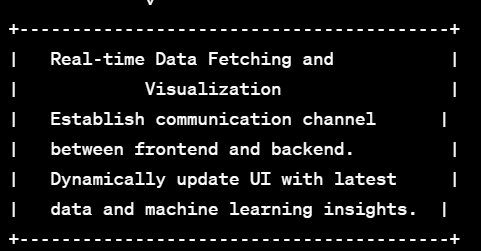
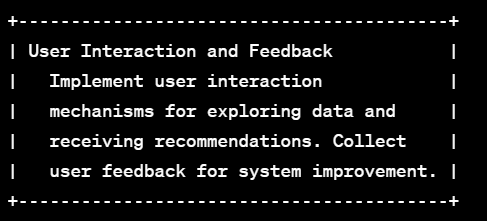
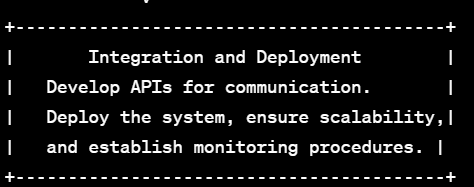
**Energy-Efficient Appliances**

Power consumption analysis for households is an ML project that involves using machine learning algorithms to analyse and predict the energy consumption patterns of residential buildings. The goal of this project is to help homeowners and utility companies better manage their energy usage, reduce waste, and lower costs.

The project involves collecting data on energy consumption and related factors such as weather, time of day, and occupancy. This data is then used to train machine learning models to make accurate predictions of future energy consumption based on these factors. The models can be used to identify patterns in energy usage and make recommendations for ways to reduce energy waste and improve efficiency.

Overall, power consumption analysis for households is an important application of machine learning that has the potential to make a significant impact on energy usage and sustainability.

**Technical Architecture:**  
 

**Project Flow:**

* User interacts with the UI to enter the input.
* Entered input is analysed by the model which is integrated.
* Once model analyses the input the prediction is showcased on the UI

To accomplish this, we have to complete all the activities listed below,

* Define Problem / Problem Understanding
  1. Specify the business problem

○ Business requirements ○ Literature Survey ○ Social or Business Impact.

* Data Collection & Preparation
  1. Collect the dataset

○ Data Preparation

* Exploratory Data Analysis
  1. Descriptive statistical

○ Visual Analysis

* Model Building
  1. Training the model in multiple algorithms

○ Testing the model

* Performance Testing & Hyperparameter Tuning
  1. Testing model with multiple evaluation metrics

○ Comparing model accuracy before & after applying hyperparameter tuning

* Model Deployment
  1. Save the best model

○ Integrate with Web Framework

* Project Demonstration & Documentation
  1. Record explanation Video for project end to end solution

○ Project Documentation-Step by step project development procedure

**Prior Knowledge:**

You must have prior knowledge of following topics to complete this project.

* ML Concepts o Supervised learning: https://www.javatpoint.com/supervised-machine-learning o

Unsupervised learning: https://www.javatpoint.com/unsupervised-machine-learning

* Decision tree: https://www.javatpoint.com/machine-learning-decision-tree-classificationalgorithm
* Random forest: <https://www.javatpoint.com/machine-learning-random-forest-algorithm>• KNN: <https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning>
* Xgboost:

https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-tounderstand-the-math-behind-bo ost/

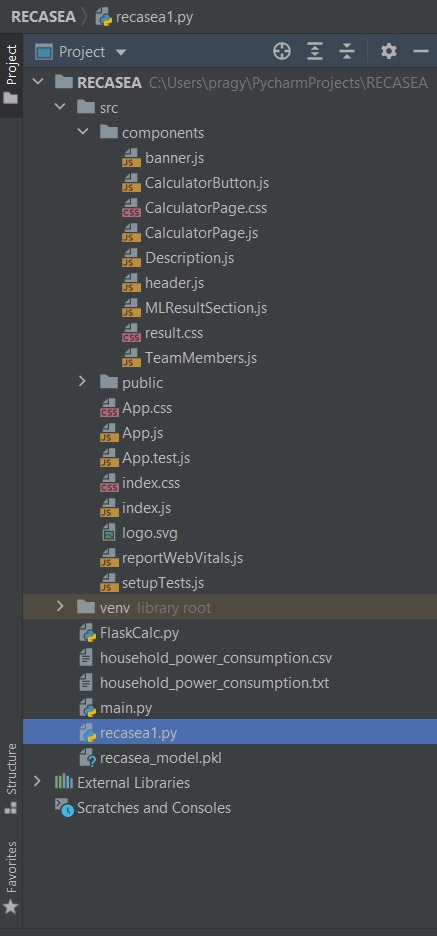
* Evaluation metrics:

https://www.analyticsvidhya.com/blog/2019/08/11-important-modelevaluation-error-metrics/

* Flask Basics: <https://www.youtube.com/watch?v=lj4I_CvBnt0>

**Project Structure:**

Create the Project folder which contains files as shown below



* + We are building a flask application which needs HTML pages stored in the templates folder and a python script app.py for scripting.
  + PCASSS\_model.pkl is our saved model. Further we will use this model for flask integration.
  + Training folder contains a model training file.
  + IBM folder contains IBM notebook and flask file also

# Milestone 1: Define Problem / Problem Understanding

**Activity 1: Specify the business problem**

Refer Project Description

## Activity 2: Business requirements

Power consumption analysis for households is an important tool that can provide valuable insights into the usage patterns of electricity in a household. Here are some business requirements that are necessary for a power consumption analysis for households:

* Accuracy: The analysis should provide accurate information on the energy consumption of each appliance and device in the household. This will help homeowners identify which appliances are consuming the most energy and make informed decisions about how to reduce their energy consumption.
* Real-time monitoring: The analysis should be able to monitor the energy consumption of the household in real-time. This will enable homeowners to make adjustments to their energy consumption on a daily basis, which can lead to significant savings on their electricity bills.

## Activity 3: Literature Survey

Power consumption analysis for households is a complex and multidisciplinary research field that involves the use of various techniques and methods. This literature survey has provided an overview of some of the recent research on this topic, including studies that use machine learning techniques, smart meters, energy audits, and simulation models. The research has identified several factors that affect the power consumption of households, such as household size, income, the type of appliances used, the energy efficiency of appliances, the size of the air-conditioner, and the usage patterns of the household.

**Activity 4: Social or Business Impact.**

Social Impact :- Promotes energy conservation: By monitoring power consumption, households can identify which appliances are consuming the most energy and take steps to reduce their usage. This promotes energy conservation and reduces carbon footprint, helping to mitigate the impact of climate change.

Business Model/Impact :- Energy companies and appliance manufacturers can use power consumption data to target consumers with tailored marketing campaigns, promoting energy-efficient appliances or incentivizing behaviour changes that reduce overall power consumption.

# Milestone 2: Data Collection & Preparation

ML depends heavily on data. It is the most crucial aspect that makes algorithm training possible. So, this section allows you to download the required dataset.

## Activity 1: Collect the dataset

There are many popular open sources for collecting the data. Eg: kaggle.com, UCI repository, etc.

In this project we have used .csv data. This data is downloaded from kaggle.com. Please refer to the link given below to download the dataset.

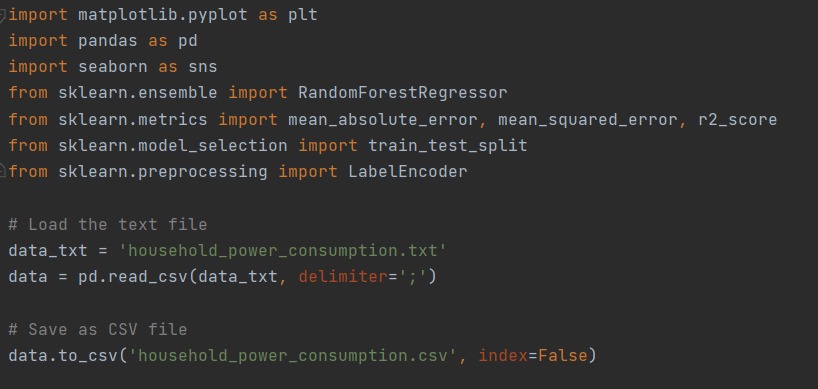
Link: [Household](https://www.kaggle.com/datasets/uciml/electric-power-consumption-data-set) [Electric](https://www.kaggle.com/datasets/uciml/electric-power-consumption-data-set) [Power](https://www.kaggle.com/datasets/uciml/electric-power-consumption-data-set) [Consumption](https://www.kaggle.com/datasets/uciml/electric-power-consumption-data-set) [|](https://www.kaggle.com/datasets/uciml/electric-power-consumption-data-set) [Kaggle](https://www.kaggle.com/datasets/uciml/electric-power-consumption-data-set)

As the dataset is downloaded. Let us read and understand the data properly with the help of some visualisation techniques and some analysing techniques.

**Note:** There are a few techniques for understanding the data. But here we have used some of it. In an additional way, you can use multiple techniques.

## Activity 1.1: Importing the libraries

Import the necessary libraries as shown in the image.



## Activity 1.2: Read the Dataset

Our dataset format might be in .csv, excel files, .txt, .json, etc. We can read the dataset with the help of pandas.

In pandas we have a function called read\_csv() to read the dataset. As a parameter we have to give the directory of the csv file.



tail() method is used to return last n (5 by default) rows of a DataFrame or series.

## Activity 2: Data Preparation

As we have understood how the data is, let's pre-process the collected data.

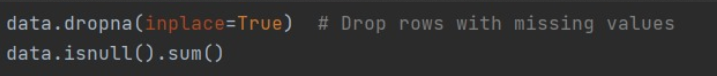
The download data set is not suitable for training the machine learning model as it might have so much randomness so we need to clean the dataset properly in order to fetch good results. This activity includes the following steps.

* Handling missing values
* Handling categorical data
* Checking descriptive statistics and correlation

Note: These are the general steps of pre-processing the data before using it for machine learning. Depending on the condition of your dataset, you may or may not have to go through all these steps.

## Activity 2.1: Handling missing values

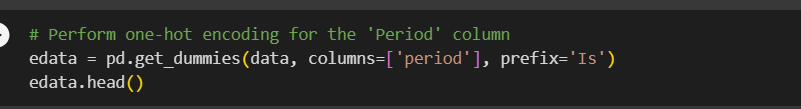
Displaying the number of rows and columns of dataset



## Activity 2.2: Handling Categorical Values

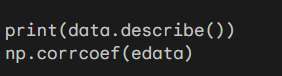
As we can see our dataset has categorical data we must convert the categorical data to integer encoding or binary encoding.

To convert the categorical features into numerical features we use encoding techniques. There are several techniques but in our project we are using manual encoding with the help of list comprehension.

● In our project, there is no categorical columns, replaced all the special characters from the columns.  
  


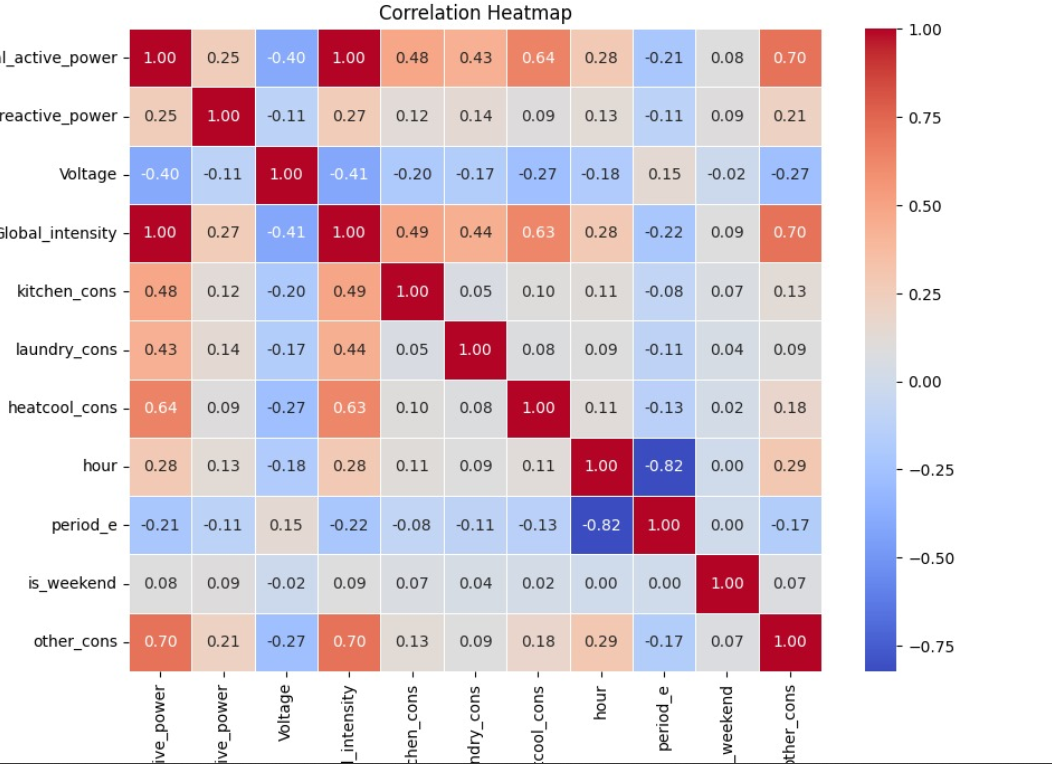
## Activity 2.3: checking descriptive statistics and correlation

**describe ()** functions are used to compute values like count, mean, standard deviation and IQR(Inter Quartile Ranges) and give a summary of numeric type data.



**corr()** is used to find the pairwise correlation of all columns in the dataframe. Any na values are automatically excluded. For any non-numeric data type columns in the dataframe it is ignored.

A heat map (or heatmap) is a graphical representation of data where values are depicted by color. Heat maps make it easy to visualize complex data and understand it at a glance. Here lighter colour means that the columns are highly correlated with our target data.



# Milestone 3: Exploratory Data Analysis

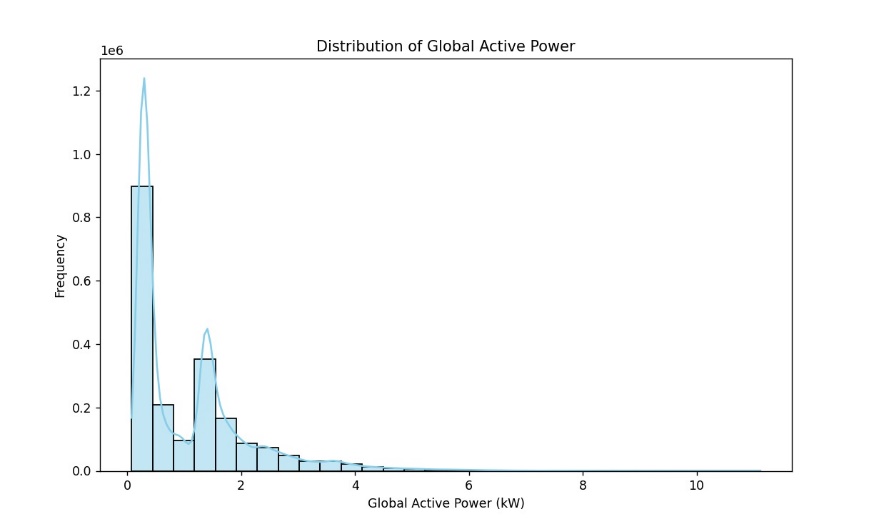
## Activity 1: Visual analysis

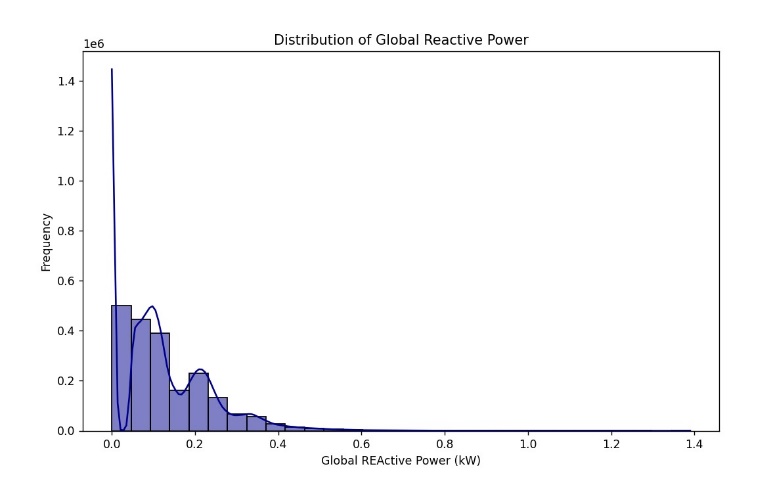
Visual analysis is the process of using visual representations, such as charts, plots, and graphs, to explore and understand data. It is a way to quickly identify patterns, trends, and outliers in the data, which can help to gain insights and make informed decisions.

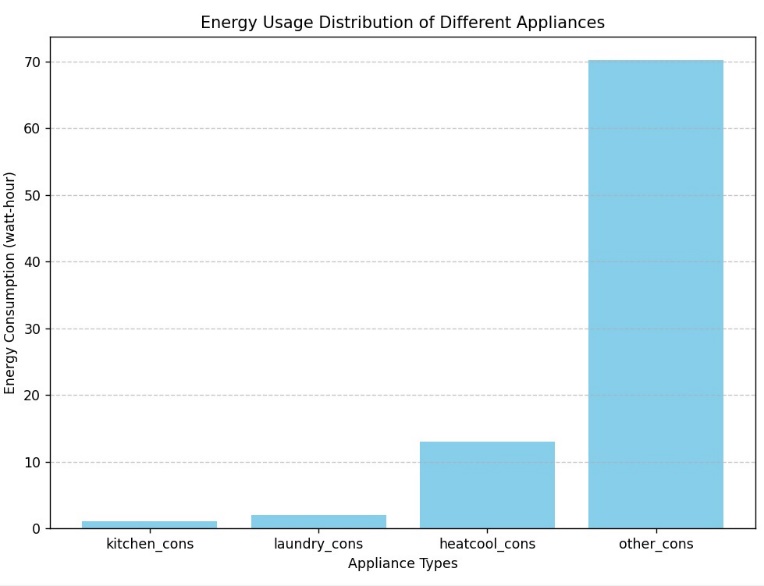
## Activity 1.1: Univariate analysis

In simple words, univariate analysis is understanding the data with single feature. Here we have displayed two different graphs such as distplot and countplot.

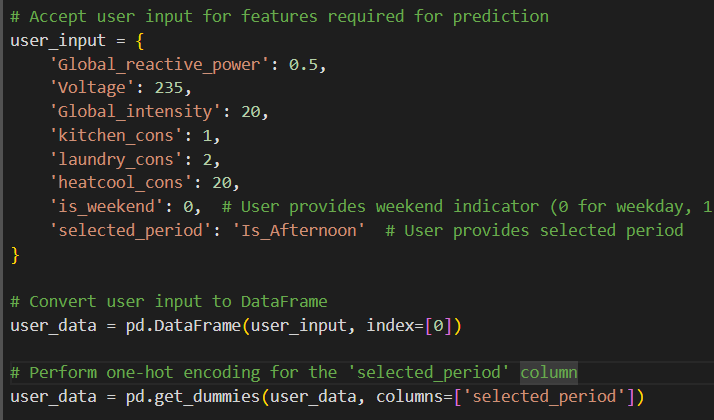
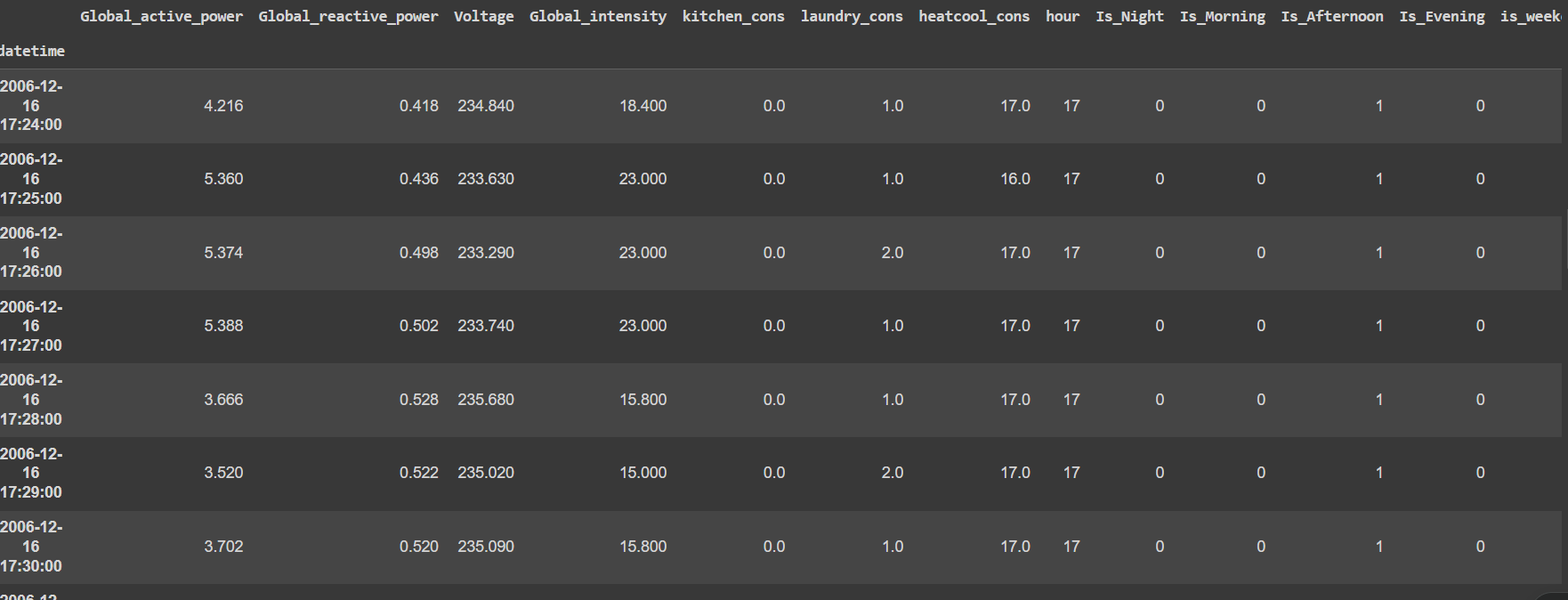
Seaborn package provides a wonderful function distplot. With the help of distplot, we can find the distribution of the feature. To make multiple graphs in a single plot, we use subplot.





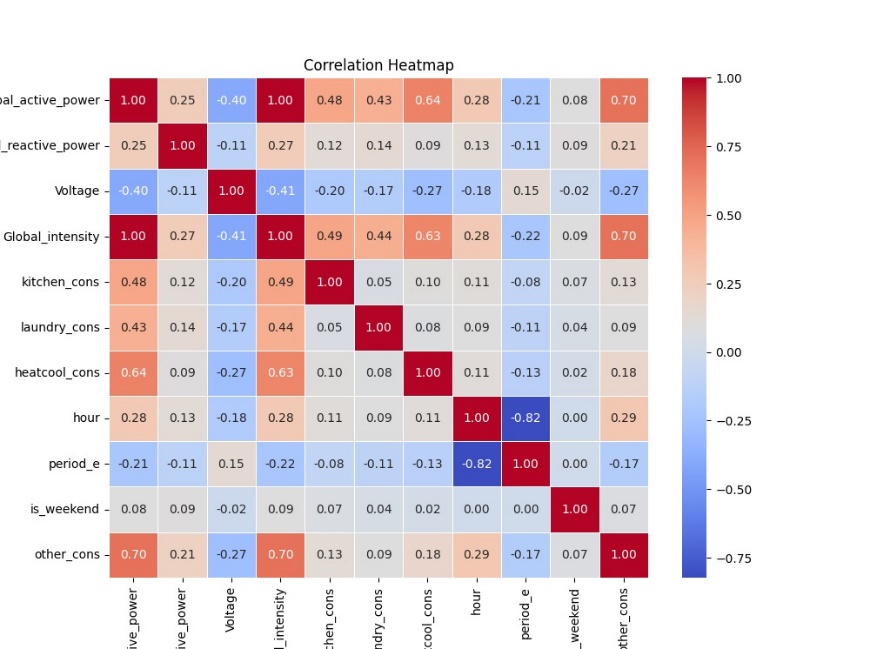


**Activity 1.2: Bivariate analysis**

Seaborn jointplot lets you show a plot of two variables with bivariate and univariate graphs.  
  
  
  


## Activity 1.3: Multivariate analysis

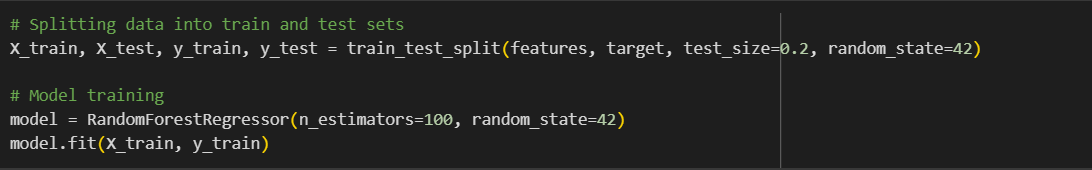
In simple words, multivariate analysis is to find the relation between multiple features. Here we have used swarmplot from seaborn package.we can consider heatmap for multivariate analysis.



**Splitting data into train and test**

Now let’s split the Dataset into train and test sets. First split the dataset into x and y and then split the data set.

Here x and y variables are created. On x variable, df is passed with dropping the target variable. And on y target variable is passed. For splitting training and testing data we are using train\_test\_split() function from sklearn. As parameters, we are passing x, y, test\_size, random\_state.



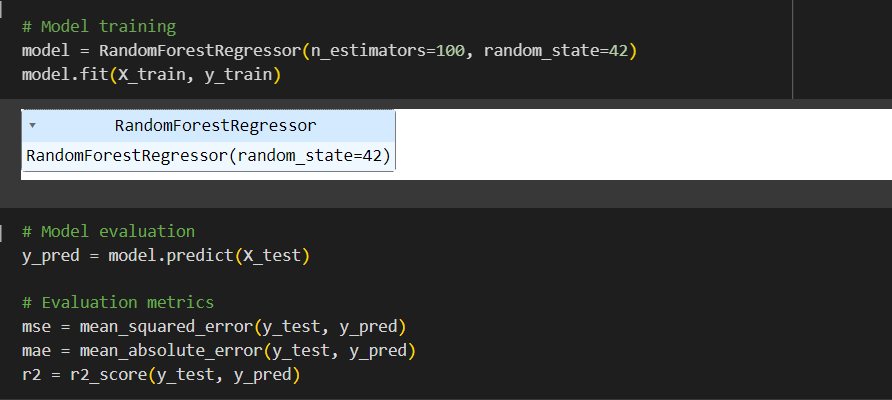
# Milestone 4: Model Building

**Activity 1: Training the model in suitable algorithms**

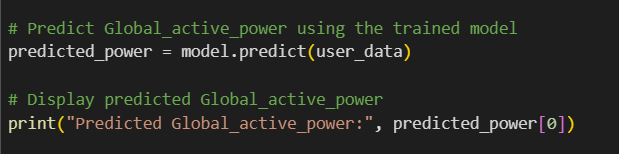
Now our data is cleaned and it’s time to build the model. We can train our data on different algorithms. For this project we are applying three classification algorithms. The best model is saved based on its performance.

## Activity 1.3: Random Forest Regressor Model

A function named randomForest is created and train and test data are passed as the parameters. Inside the function, RandomForestRegressor algorithm is initialised and training data is passed to the model with .fit() function. Test data is predicted with. predict() function and saved in a new variable. For evaluating the model, r2score, MSE.RMSE,MAE.



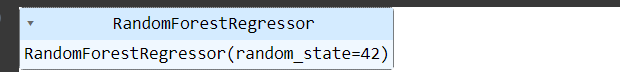
### Activity 2: Testing the models

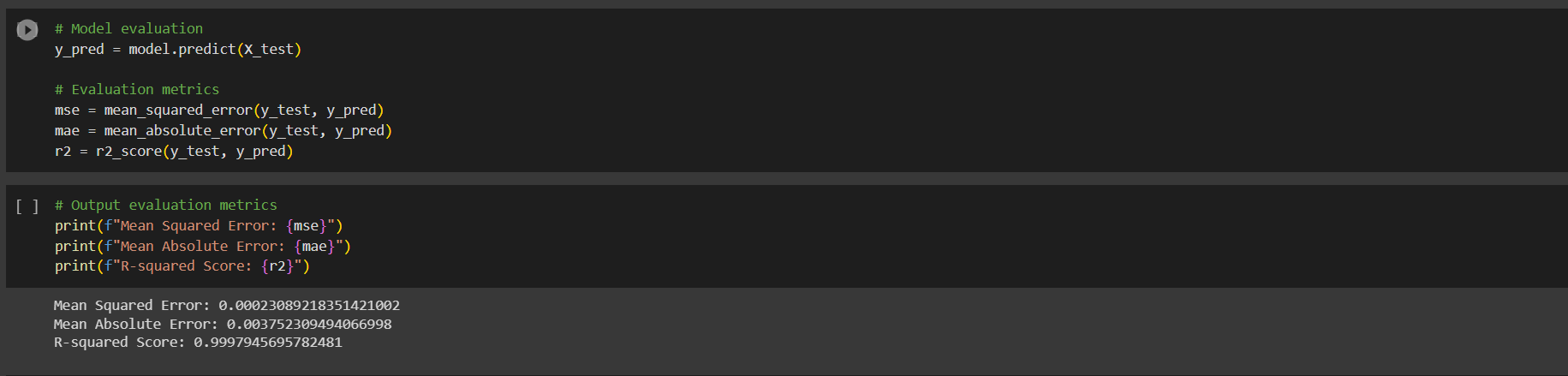


# Milestone 5: Performance Testing & Hyperparameter Tuning

## Activity 1: Testing model with multiple evaluation metrics

Multiple evaluation metrics means evaluating the model's performance on a test set using different performance measures. This can provide a more comprehensive understanding of the model's strengths and weaknesses. We are using evaluation metrics for regression tasks including MAE, MSE, RMSE and R Square

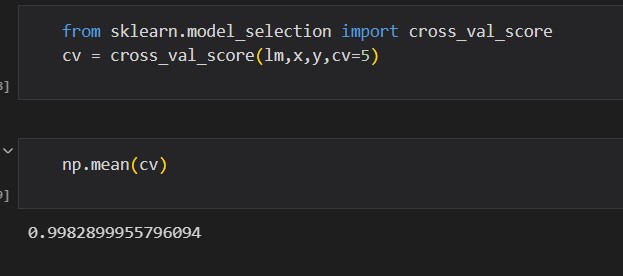
**Activity 1.1: Compare the model**



**Activity 2: Comparing model accuracy before & after applying hyperparameter tuning (Hyperparameter tuning is optional. For this project it is not required.)**

Evaluating performance of the model From sklearn, cross\_val\_score is used to evaluate the score of the model. On the parameters, we have given lm(model name), x, y, cv (as 5 folds). Our model is performing well.

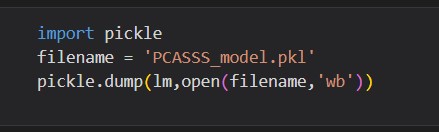
**Note:** To understand cross validation, refer to this [link](https://towardsdatascience.com/cross-validation-explained-evaluating-estimator-performance-e51e5430ff85.)



# Milestone 6: Model Deployment

## Activity 1: Save the best model

Saving the best model after comparing its performance using different evaluation metrics means selecting the model with the highest performance and saving its weights and configuration. This can be useful in avoiding the need to retrain the model every time it is needed and also to be able to use it in the future.



## Activity 2: Integrate with Web Framework

In this section, we will be building a web application that is integrated to the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.

This section has the following tasks

* Building HTML Pages
* Building server-side script
* Run the web application

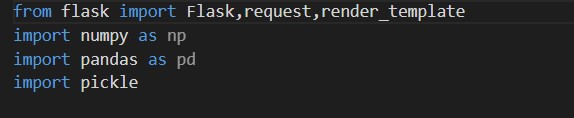
**Activity 2.1: Building Html Pages:**

For this project create two HTML files namely

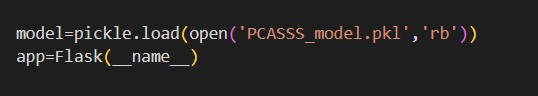
* index.html
* inspect.html
* output.html and save them in the templates folder. Refer this [link](https://drive.google.com/drive/folders/1K-C2uvRstV8x6bwsOY_BWBvXdzjbbfPF?usp=share_link) for templates.

**Activity 2.2: Build Python code:**

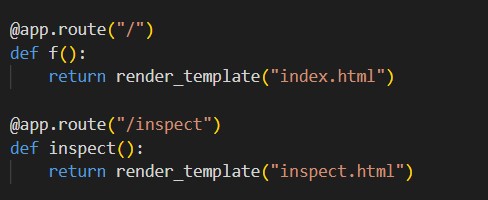
Import the libraries



Load the saved model. Importing the flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of the current module (\_\_name\_\_) as argument.



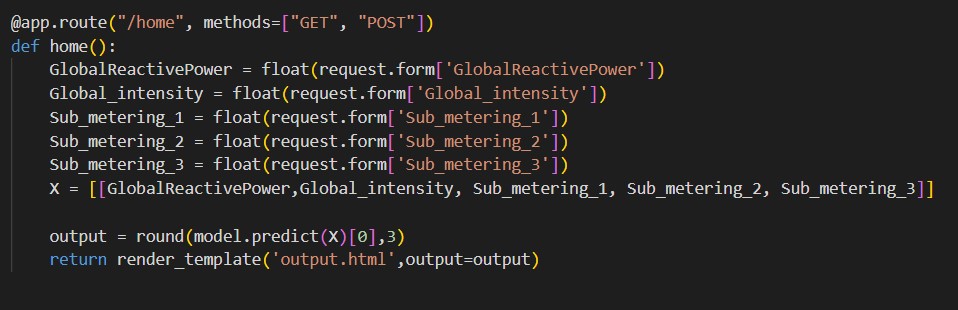
Render HTML page:



Here we will be using a declared constructor to route to the HTML page which we have created earlier.

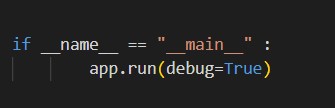
In the above example, ‘/’ URL is bound with the home.html function. Hence, when the home page of the web server is opened in the browser, the html page will be rendered. Whenever you enter the values from the html page the values can be retrieved using POST Method.

Retrieves the value from UI:



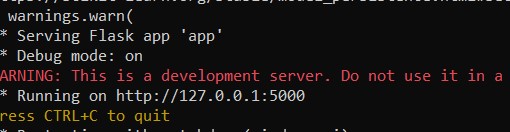
Here we are routing our app to home() function. This function retrieves all the values from the HTML page using Post request. That is stored in an array. This array is passed to the model.predict() function. This function returns the prediction. And this prediction value will be rendered to the text that we have mentioned in the output.html page earlier.

Main Function:

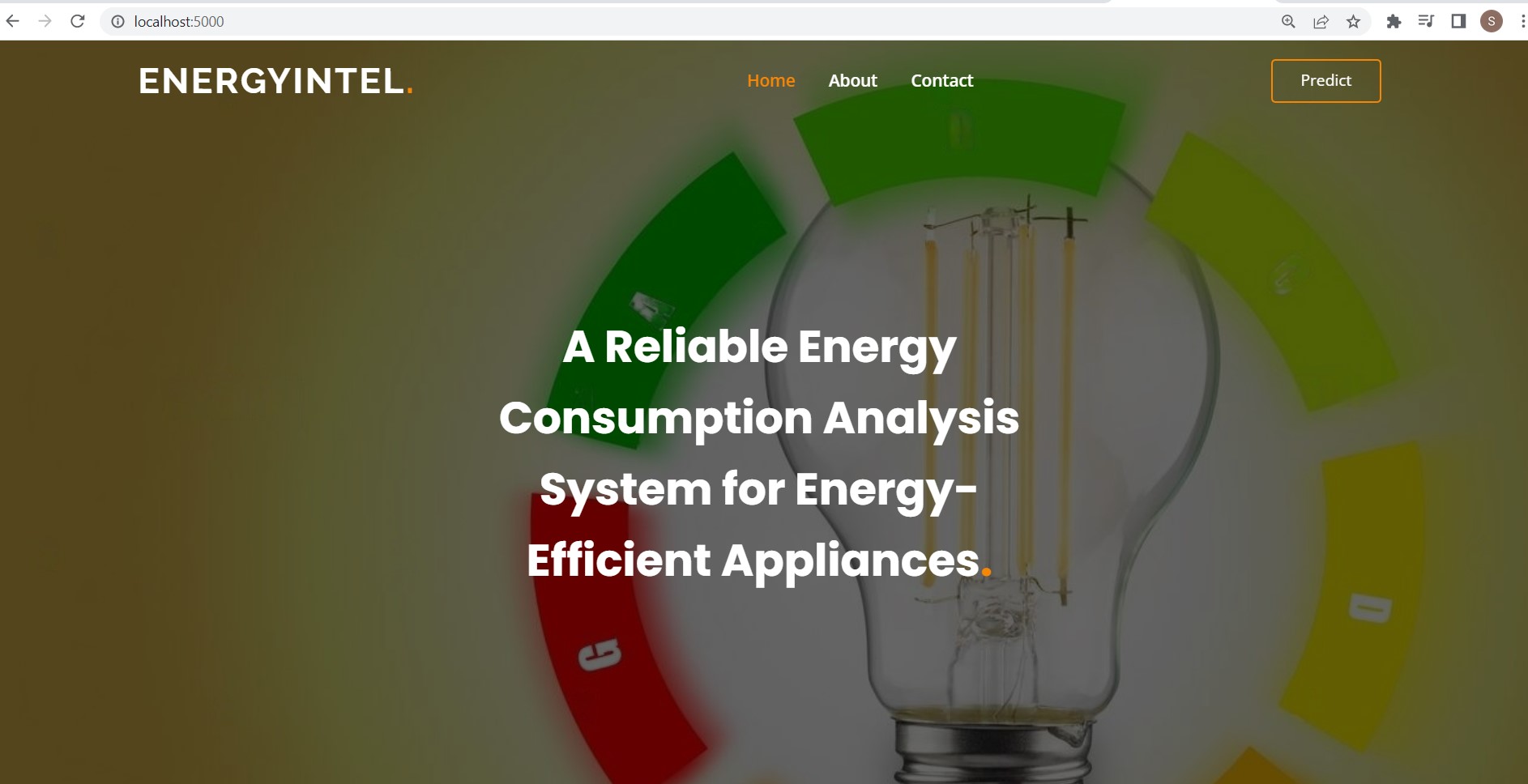


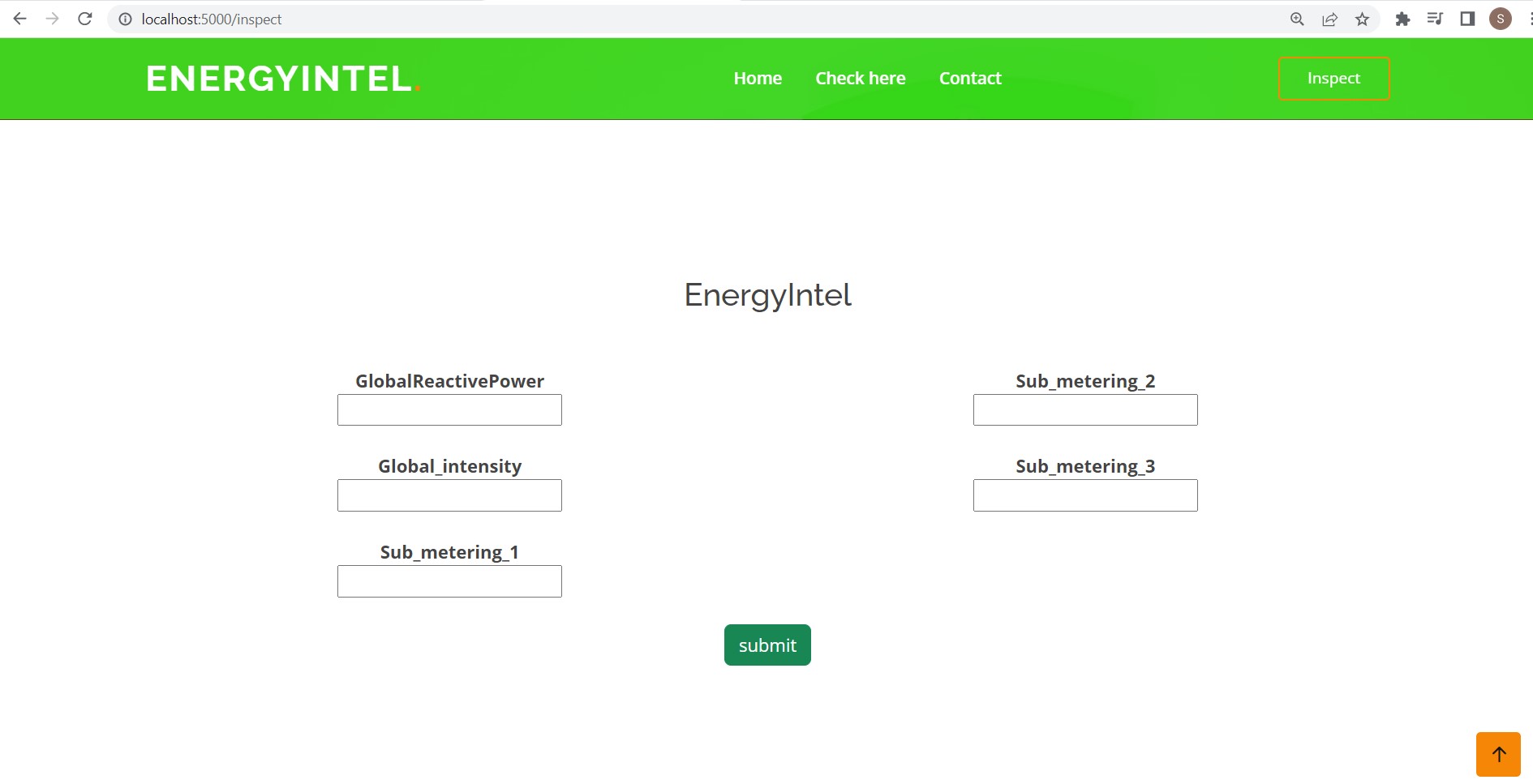
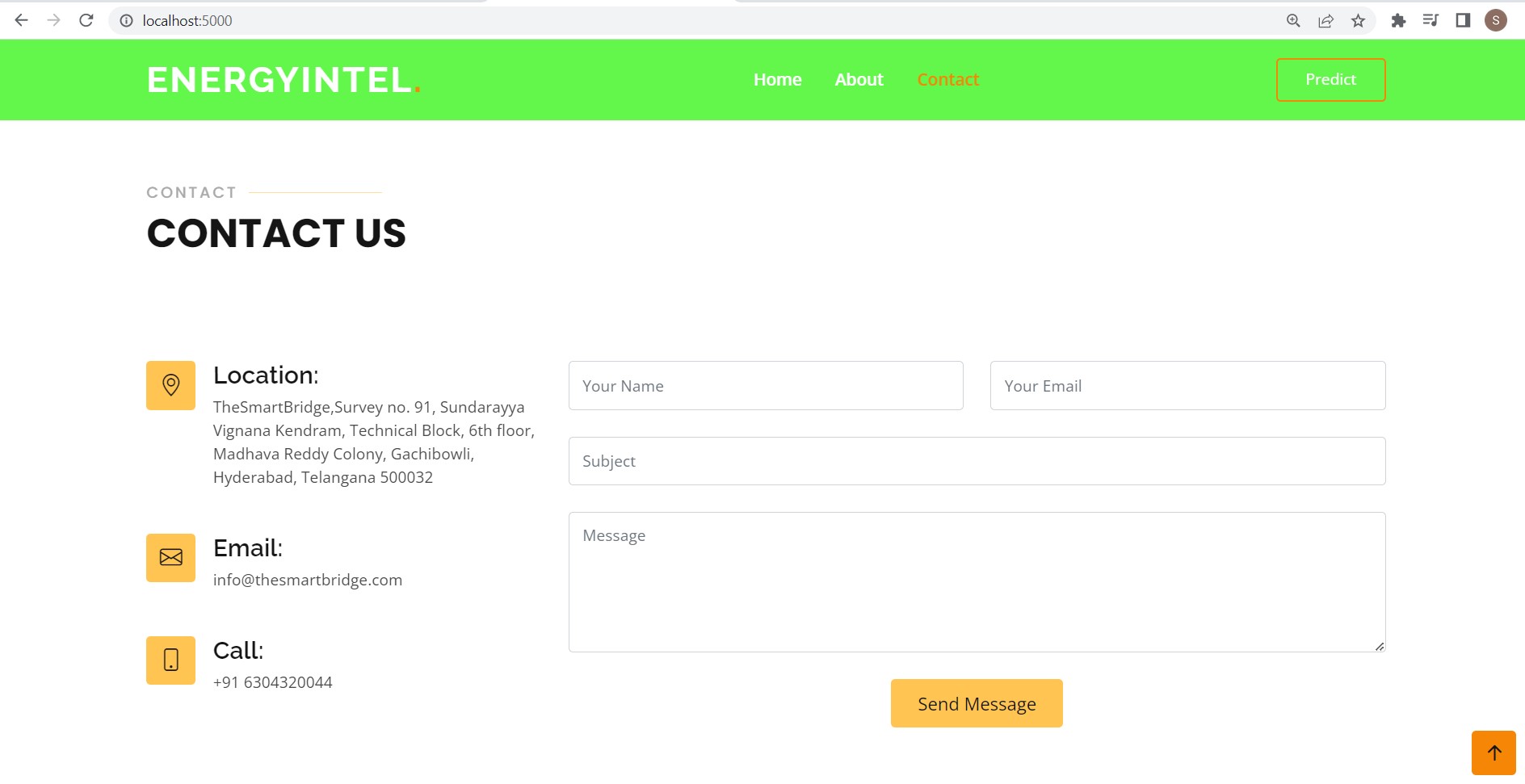
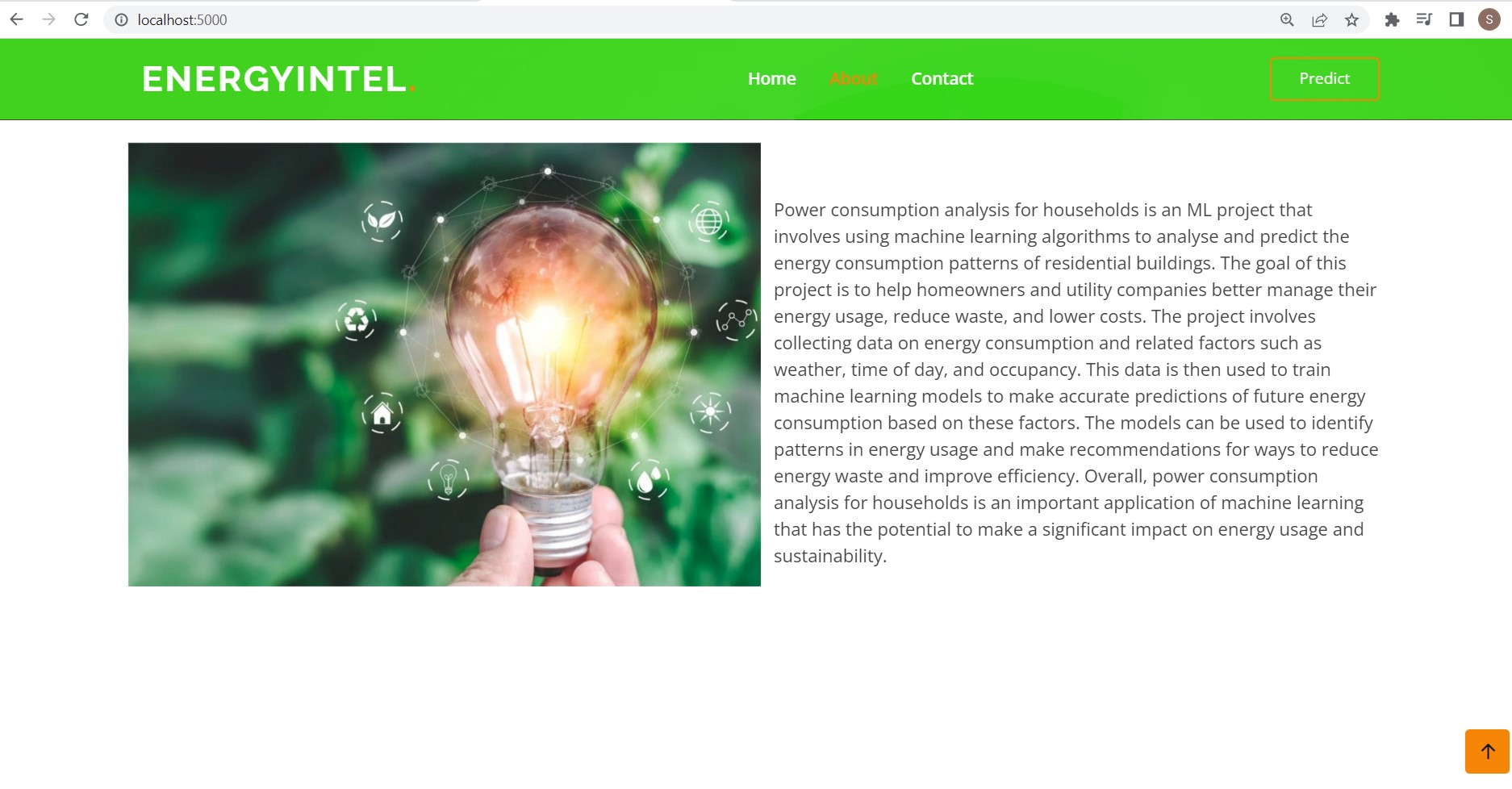
## Activity 2.3: Run the web application

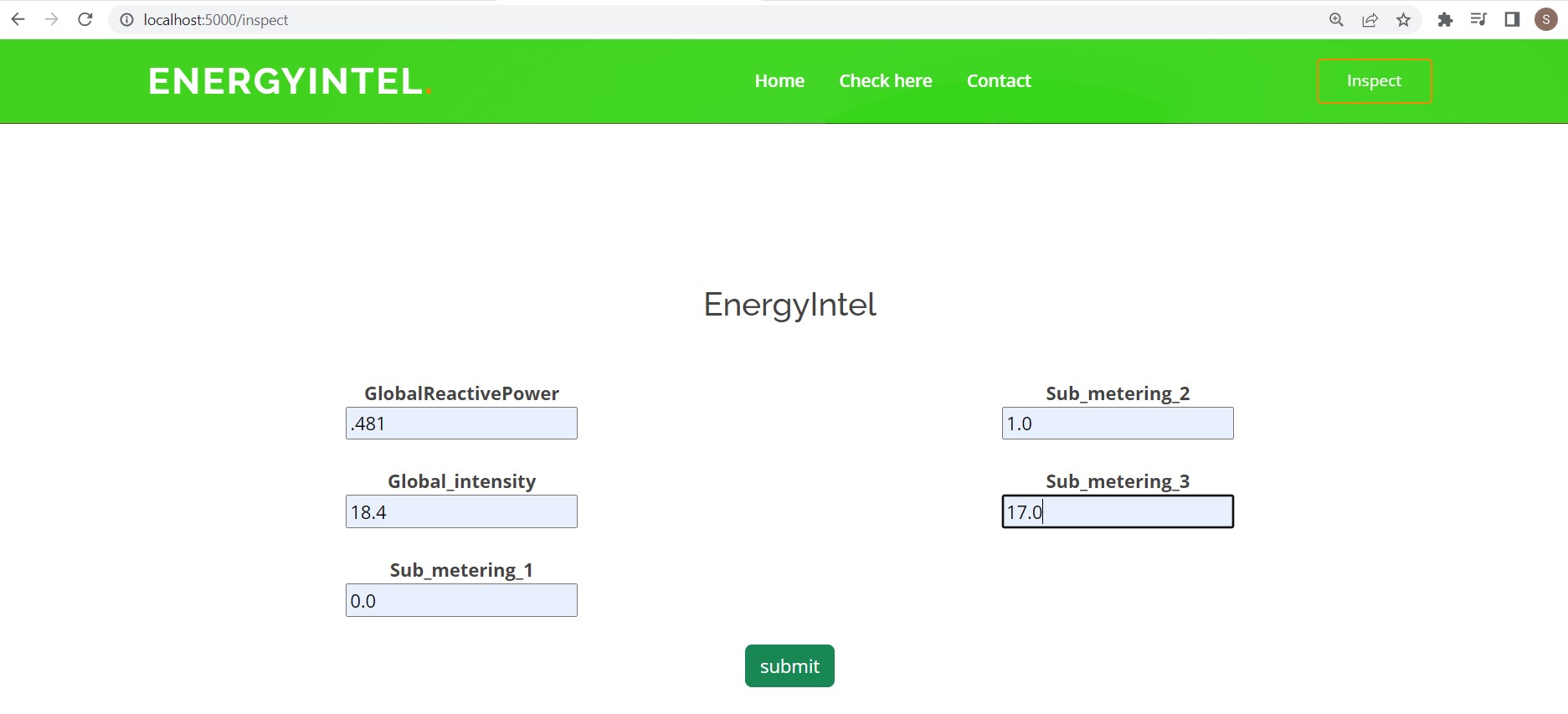
* Open anaconda prompt from the start menu
* Navigate to the folder where your python script is.
* Now type “python app.py” command
* Navigate to the localhost where you can view your web page.
* Click on the predict button from the top left corner, enter the inputs, click on the submit button, and see the result/prediction on the web.

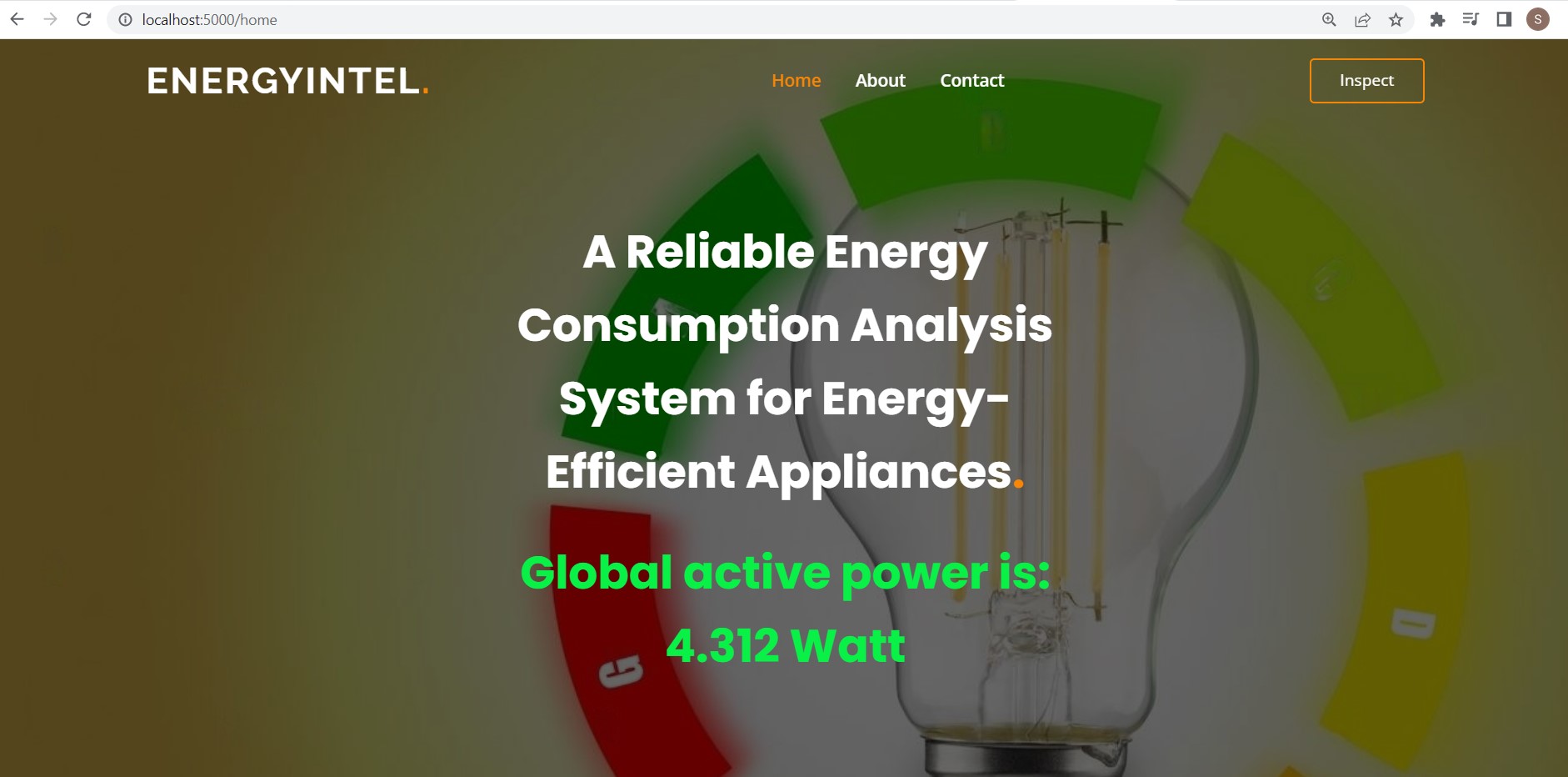


Now, Go the web browser and write the localhost URL (http://127.0.0.1:5000) to get the below result









Finally, total power consumption by all the appliances is calculated and displayed.

**Milestone 7: Project Demonstration & Documentation**

Below mentioned deliverables to be submitted along with other deliverables

**Activity 1:- Record explanation Video for project end to end solution**

## Activity 2: - Project Documentation-Step by step project development procedure

Create document as per the template provided