

# **Final Project Part 3**

## **Daily Website Traffic Trends: A Time Series Analysis**

Group 9 Team Members:

1. Priya Kumari ([kumarip@iu.edu](mailto:kumarip@iu.edu))
2. Ayesha Tajammul Ahmed Mulla ([amulla@iu.edu](mailto:amulla@iu.edu))
3. Aazin Asif Shaikh ([aazshaik@iu.edu](mailto:aazshaik@iu.edu))

## **Time Series Application Design**

### **Part 1. Time Series Application Architecture**

**Model:** We are using data stored in csv file and processing it using pandas while keeping it in the form of a dataframe.

**View:** We are using HTML, CSS and JS for the front end.

**Controller:** We are using Flask for the back end which is essentially the brain of our application performing all the connections and allowing deployment.

#### **1. Data Storage:**

The data we use is sourced as a csv file from publicly available website data. This data is then processed using the pandas library in Python and we are further creating our models by using the data stored as a pandas dataframe. We use this stored data to perform forecasting and building our time-series models.

#### **2. Language used:**

The back-end system will predominantly be built using the Python programming language.

#### **3. Data Access and Security:**

Our web application makes use of backend technologies such as Python and Flask to handle data retrieval, processing, and analysis. Data may be temporarily stored in memory or in a database for efficient access and processing. We will be using Flask-Login which is a popular extension for Flask that provides user session management and authentication features. It allows developers to manage user authentication using various strategies such as local authentication.

The web application includes a home page with a description, visualisation page for interactive visualisations, and forecasting for upcoming days. The forecasting feature and the page showing forecasting plot for a given user input will be accessible only if

a user signs up or logs in. However, the plots showing our overall data and corresponding visualization will be publicly available.

#### **4. Front-end layout Creation:**

To create the front-end layout of a website or web application, we will typically be using HTML, CSS, and JavaScript.

#### **5. Application Deployment:**

As of now, we have considered hosting it on PythonAnywhere.

#### **6. Application Interactivity:**

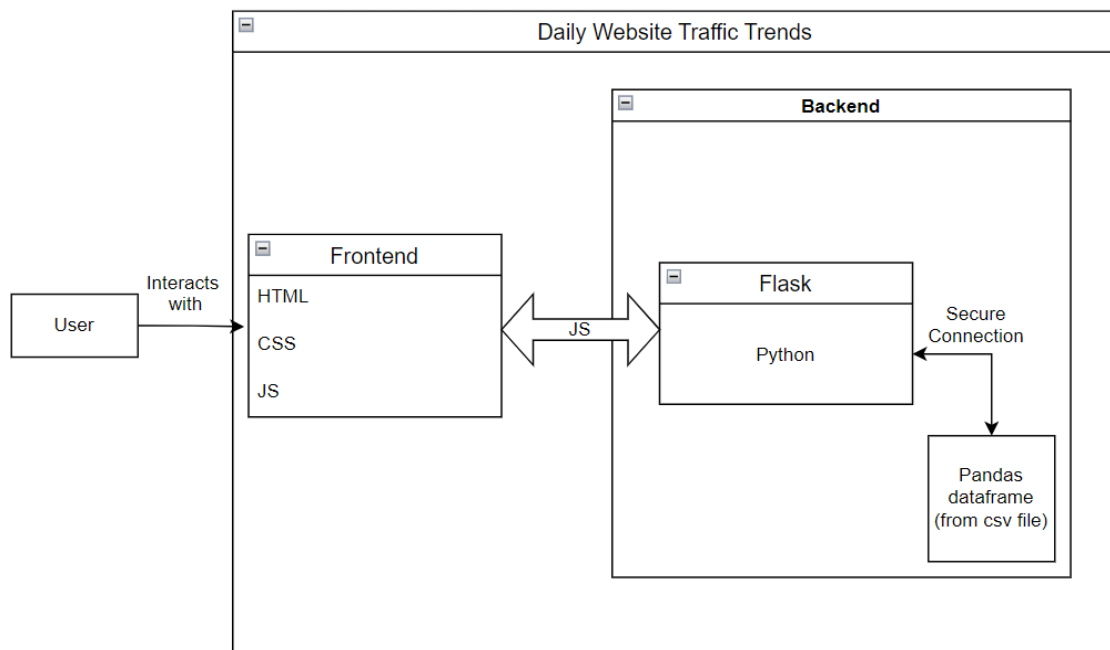
To add interactivity to our app, we are implementing a user-friendly interface that allows users to select the number of days they want to see the forecast for daily unique website visits. This selection will be made through interactive elements such as input fields.

Once the user selects the desired number of days, the app will generate a table displaying the forecasted data for website visits on a daily basis. The table will be presented in a visually appealing format, making it easy for users to understand the data.

Additionally, we will provide a graph view option that allows users to visualise the forecasted data in a graphical format. This is achieved using Plotly, which enables users to zoom in, download, and perform various actions on the graph, such as zooming in, panning, and downloading the graph.

By incorporating these interactive features, users will be able to easily interact with the app, customise their viewing preferences, and perform actions on the data to gain insights and make informed decisions.

## 7. Web App Architecture Schema.



## Part 2. App Layout

We have designed page layouts for our web application using Figma, primarily utilizing a color scheme of black, white, and orange. The layouts are carefully crafted to create a visually appealing and user-friendly interface. The key pages in our web application include:

### 1. Landing Page (Home Page)

The landing page serves as the home page of our web application, where users can learn about the application and its usefulness. The landing page consists of two buttons: sign-up and login. The sign-up button is for first-time visitors who need to register and create an account for personalized console and secure access. The login button is for returning users who can log in with their credentials to access the application.

### 2. Data Exploration and Visualization Page

Clicking the "Get Started" button on the landing page directs the user to the Data Exploration and Visualization page. This page is an interactive dashboard that displays unique visit plots showing interesting analysis of the data. It also includes a feature for forecasting the expected unique user visits in the future.

### 3. Regression Analysis and Forecasting Page

The Regression Analysis and Forecasting page is where users can input the number of days for which they want to forecast the unique visits. Clicking on the "Forecast" button displays a tabular view of data, showing the unique visit values at the corresponding dates. If users do not want to go through every row in the table, there is another feature available.

### 4. Live Forecast Plot Page

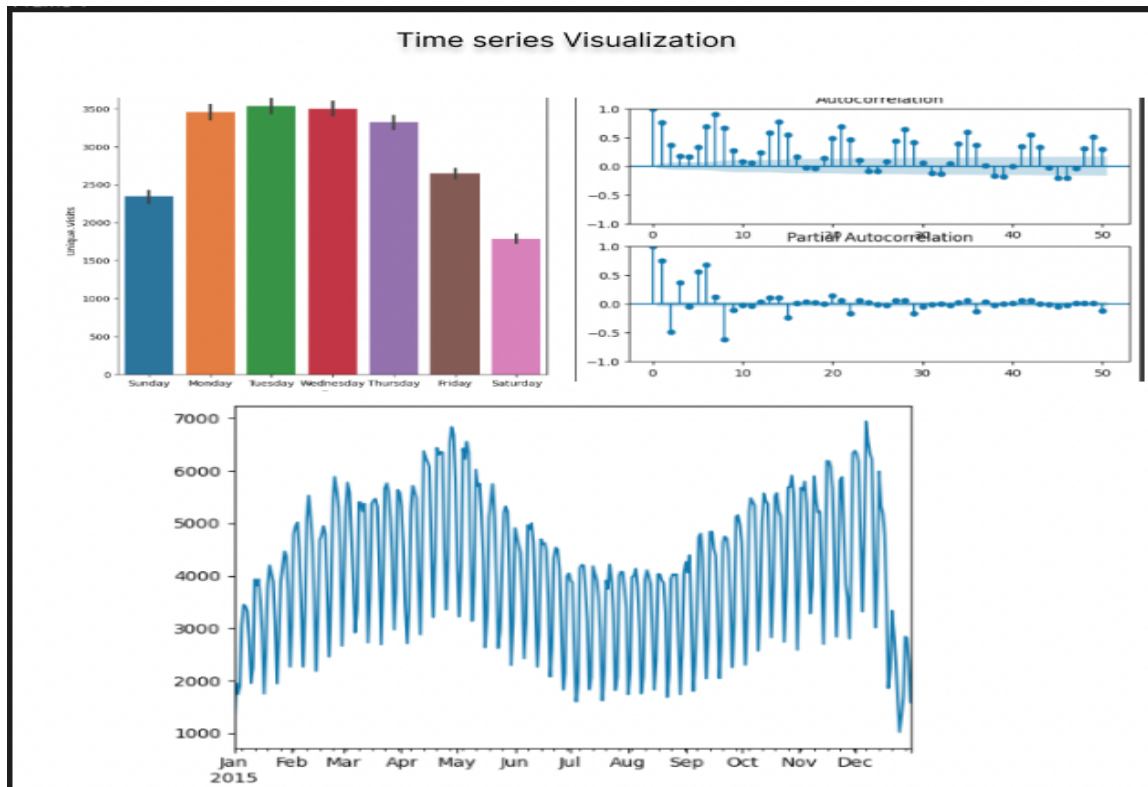
The live forecast plot is an interactive visual plot of the predicted data that users can play with and perform detailed analysis. This is available as a separate page.

Overall, our web application offers a landing page with sign-up and login buttons, a data exploration and visualization page with forecasting feature, a regression analysis and forecasting page with tabular view, and a separate live forecast plot page for detailed analysis.

## Home Page



## Time Series Visualisation page



## Regression Analysis and Forecasting Page Layout

Desktop - 1

# Regression Analysis and Forecasting

Forecast the Number of Unique Visits

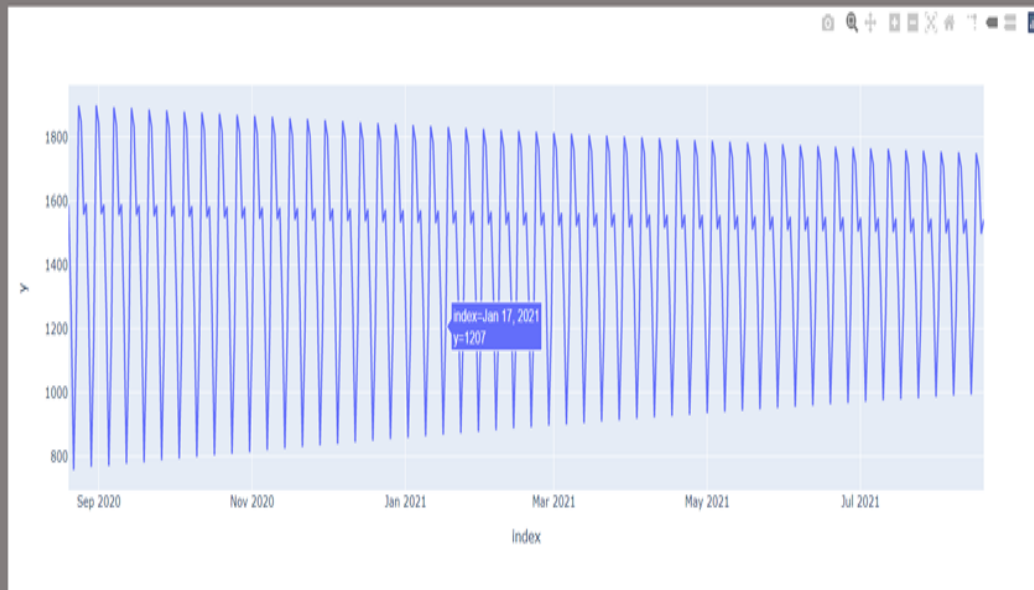
**Days**

Enter number of days to forecast

FORECAST

LIVE FORECAST PLOT

## Live Forecast Plot Page



### Part 3. Team Work

Team Member	Tasks	Description
<p>Ayesha Tajammul Ahmed Mulla <a href="mailto:amulla@iu.edu">amulla@iu.edu</a></p>	<p>Designing app layout using Figma.</p> <p>Designing the login feature on the Homepage.</p> <p>Designing Forecasting and Regression Analysis webpage</p>	<ol style="list-style-type: none"> <li>1. Designing the layout of the web pages such as the home page, Sign-up and Login Page, Data Visualization Dashboard and Forecasting Page using Figma tool.</li> <li>2. Create a login page enabling registered users to login to the website with secure authentication using HTML, CSS, JavaScript</li> <li>3. Create a forecasting webpage using Python, Flask, HTML, CSS, JS to perform predictions using the ARIMA model selected in the previous phase on future data where User can input "n" number of days for which they want to analyze prediction along with interactive plot created using Plotly.</li> </ol>
<p>Aazin Asif Shaikh <a href="mailto:aazshaik@iu.edu">aazshaik@iu.edu</a></p>	<p>Designing the web app architecture using draw.io</p> <p>Designing the Homepage and the sign-up feature with user authentication</p> <p>Integrating the webpages using Flask</p>	<ol style="list-style-type: none"> <li>1. Design the end-to-end workflow architecture of the web app.</li> <li>2. Design a user-friendly Homepage with a sign-up feature to register a new user in the system with secure authentication using HTML, CSS, JavaScript</li> <li>3. Combining all the web pages created using Flask, Python and HTML</li> </ol>
<p>Priya Kumari <a href="mailto:kumarip@iu.edu">kumarip@iu.edu</a></p>	<p>Designing the Data Exploration and Visualisation Page</p> <p>Deployment of the web application on cloud server</p> <p>Designing the MVC Schema based on questions mentioned.</p>	<ol style="list-style-type: none"> <li>1. Design a webpage cum Dashboard to display interactive plots using Plotly to give user a visual overview of the data, its trends, stationarity and distribution.</li> <li>2. Deploying the web application using PythonAnywhere to migrate local access of the application on cloud for public accessibility.</li> <li>3. Brainstormed on the tools suitable to build an end-to-end web application and created layout using Figma.</li> </ol>