International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-12 Issue-7, June 2023

Development of a Data Analysis Module

Atharva Gupte, Kaushal Bhide, Swarupa Deshpande, Manas Apte, Rohan Rasane

Abstract*: *A module that gathers data from various sources like SQL databases or CSVs and then, with the help of this data, provides meaningful insights to the customer. The software will be an efficient way to track data from varied sources in real time. It will provide a centralized system to monitor and analyze the performance of any business/organization. With the help of this software, or dashboard an environment can be created for business analysis as well as management. This module includes all phases of the Data Analysis life cycle, like data collection, data pre-processing, data analysis, visualization, and eventually effective decision making. A holistic solution for each step is given by the software so as to yield as many insights as possible. In today's time where data is the new currency, this software or module or dashboard will provide users with a wide range of options to work with and around data. With the help of this software, one can achieve effective monitoring and evaluation of the business sector. Due to the lack of such software, the available raw data is often not transformed and used for decision-making. To fill this void, this module will play a vital role. The software will provide customers with the options to check factors like tracking progress towards a set target, effective decision-making for planning, and predicting sector trends and performances. The customers will have autonomy to work with variable sizes and types of data. Propagating the results is also an important thing for the customers; therefore, the module or dashboard provides effective data visualization tools as well. Thus, this software is defined as an end-to-end solution for the customers.

Keywords: Model Building, Data Collection, Data Analysis, Data Pre-processing, Extract, Transform, Load, SQL Database, API, DLL, NoSQL, Data Aggregation, OOP, Script Generation.

I. INTRODUCTION

In today's ever-growing world, we are surrounded by data everywhere. This data has been collected from various sources, including over-the-top platforms and social media websites, for quite some time now. Recently, with cloud computing as an emerging technology, data stored on cloud, or virtual, servers are also increasing day by day.

Manuscript received on 26 May 2023 | Revised Manuscript received on 06 June 2023 | Manuscript Accepted on 15 June 2023 | Manuscript published on 30 June 2023.

*Correspondence Author(s)

Atharva Gupte*, Department of Computer Engineering, Marathwada Mitra Mandal College of Engineering, Pune (Maharashtra) India. atharvagupteag143@gmail.com, ORCID ID: 0009-0004-6054-5923

Kaushal Bhide, Department of Computer Engineering, Marathwada Mitra Mandal College of Engineering, Pune (Maharashtra) India. kbhide79@gmail.com, ORCID ID: 0009-0005-4451-8076

Swarupa Deshpande, Department of Computer Engineering, Marathwada Mitra Mandal College of Engineering, Pune (Maharashtra) India. swarupadeshpande@mmcoe.edu.in, ORCID ID: 0009-0006-3711-6420

Manas Apte, Department of Computer Engineering, Marathwada Mitra Mandal College of Engineering, Pune (Maharashtra) India. aptemanas01@gmail.com, ORCID ID: 0009-0001-1775-7152

Rohan Rasane, Department of Computer Engineering, Marathwada Mitra Mandal College of Engineering, Pune (Maharashtra) India. rohan12rasane@gmail.com, ORCID ID: 0009-0004-0779-3514

With the advent of technology, IoT devices are booming in the market for the convenience they offer. Moreover, with big data at hand, the need to process this data grows by the second. Technological advancements like the integration of the IoT and cloud pose the challenge of successful data analysis. Big data refers to the concept of large volumes of data that are generated via many sources, like social media or the Internet of Things. These 'things' perform communications between a collective network. IoT aims to do tasks "smartly!". They are used to reduce human efforts and minimize their intervention. The range of applications for these smart devices is huge, from daily usage in our homes to factories and automobiles. The project focuses on analyzing the data gathered through these various sources and devices in the form of SQL or CSVs.

After this initial and essential step, we focus on efficiently handling the data gathered through the infrastructure. Various ETL libraries need to be explored for this, given that to operate on any data, ETL forms a prominent part of the process. Once we have gathered the data from multiple sources, we focus on performing various tasks on it. These operations include pivoting, joining, and concatenating.

Data analysis is the process of systematically applying statistical and/or logical techniques to describe and illustrate, condense and recap, and evaluate data. While data analysis in qualitative research can include statistical procedures, many times analysis becomes an ongoing iterative process where data is continuously collected and analyzed almost simultaneously. To analyze this data and cater to users' requirements, we chose to develop a data analysis platform. Development of a data analysis module that can provide every step in a data analysis life cycle. There is no product on the market that provides an end-to-end solution to the customer. This module will help the customers work with their data within the organization.

The project is the development of a module that is inclusive of all data analysis steps and provides effective decision making (model building), all in one. This module includes phases like data collection, data pre-processing, data analysis, visualization, and, eventually, decision making with the help of machine learning. The software required for the development includes Python, C#, Angular, the .Net framework etc. This module will be useful for enhancing businesses and will cater to every data centric need. Our team is currently working on a data analysis and visualization platform that can accept data from sources like CSV, SQL databases, APIs, etc. It will have a web-based user interface. It will have the ability to translate the insights from one analysis into another. Users will also be able to share the results with other relevant users.

II. LITERATURE SURVEY

Jakub Swacha et al., Python and C#: A Comparative Analysis from Students' Perspective [2], In a survey of 12 questions, C# falls behind when compared to Python. The comparison between these two languages is done based on criteria of simplicity at first contact, easier learning, counter intuitiveness, debugging difficulty, faster programming, shorter source code, readable source code, convenience with respect to the development environment, easy set-up and usage at home, build speed and execution speed, and finally its professional and nonprofessional aspects. Python offers a variety of ETL libraries like Luigi, Bonobo, or even Apache Spark. But Pandas are the most highly ranked amongst them and provide stability [1]. Finally, the paper concludes that Python is more dominant than C# but also clarifies that a more detailed analysis might yield different results.

Vincent C. Emeakaroha et al., 'A Cloud-Based IoT Data Gathering and Processing Platform' [3], focus on data gathering from the cloud. It first reviews the state-of-the-art embodied by the existing solutions and discusses their strengths and weaknesses. It cites how the applications of IoT or cloud technologies have shown exponential growth in commercial value. It fits best with their objective because a cloud-based data integration and analysis module becomes more significant in real-world usage by the minute. Inspired by this fact, more generalized applications for sensors and smart devices are becoming significant in real-world usage. The gathering, storage, and processing of such locally generated data is becoming impractical due to the number of sensors and the volume of data they generate. Hence, an over-the-top data analysis module.

In Dr. S. Ramakrishna et al., A Study of Extract-Transform-Load (ETL) Processes [1], this paper discusses the ETL processes in detail. While working with data, ETL forms an integral part of analysis. It is the integration layer in a data warehousing environment. ETL tools pull data from several sources (database tables, flat files, ERP, the internet, and so on). This is the extraction phase. This extracted data is subjected to complex operations and amendments, like joining or pivoting, in the transform layer. Finally, in the end, data is loaded into the target, which is a data warehouse store in the DW environment, the Load layer. The paper focuses on open-source and commercially available ETL tools. It also goes on to discuss the drawbacks and issues one might encounter while operating an ETL tool. To sum it up, it gives insights about ETL as a tool and the process itself.

Nikos Fotiou et al., Smart IoT Data Collection [4], aim to develop and experiment with procedures for efficiently collecting IoT data while achieving target requirements. It elaborates in terms of data accuracy, timeliness, energy efficiency, and privacy protection. The paper has an approach to processing huge amounts of data generated through sensors, signaling, and communications. It filters the IOT devices based on different requirements. These are sorted rather than filtered based on their precision, time delay, and energy usage. The paper then condenses its approach to accuracy-driven data collection. It also considers privacy-based data collection but concludes that the noise added to the analysis process has a negligible effect on overall statistics. The experiments studied conclude that additive increase or multiplicative decrease The procedure is robust for the calculation of various types of data. These cover climate phenomena and weather measurements.

Teo Susnjak et al., A Learning Analytics Dashboard: A Tool for Providing Actionable Insights to Learners [5], discuss that analytics is divided into sectors such as prescriptive, descriptive, or predictive analysis. It focuses more on analyzing the dashboard. This means that the aim is to understand and study the analysis platform, the tools it offers, or the services it provides. The authors comprehensively surveyed existing LADs and analyzed them through the prism of the sophistication of the insights they deliver and the ways in which they help learners make informed decisions about making adjustments to their learning habits.

III. OBJECTIVES

The integration of big data with the cloud poses the challenge of successful data analysis. To cater to the huge amount of data being generated by IoT devices, we aim to build a data analysis platform. The platform would be a one-stop solution for all types of analysis. The objective is to provide the user with a packaged deal, including convenience and budget-friendliness, that is free of cost. We focus on extracting user or device data from not one but multiple sources. This is done by loading it into data frames. Various ETL libraries need to be explored for this, given that to operate on any data, ETL forms a prominent part of the process. A complete analysis package that could effectively handle all the business needs of clients seamlessly and provide enough help so as to grow their business. As the developers, our only objective is that all the clients that resort to our dashboard get all their requirements satisfied and are happy with the analysis they have as they had intended.

IV. METHODOLOGY

Since our goal is to build a data analysis dashboard, we start with the most important thing, which is the backend, which can be considered the heart of our system. Initially, we defined what types of data sources we were going to define and let users use them. After defining the types, we build the basic programming infrastructure to store and use the data of the respective source types.

The three most basic steps of data analysis are ETL (Extract Load Transform). So, we now start with the extract process, wherein we get the data inputted by the user and store it in backend data structures. Then we let the user perform basic pre-processing on the raw data, such as removing unnecessary columns, rows, etc. After these steps are done, we then need to update the stored data in the backend with the user's choice data, which is at the frontend. After that is done, we will now be building the infrastructure for performing transformations on that data, such as pivot, sum, different joins, aggregate, average, etc. All these infrastructures are in OOP form as classes and objects. Once all the required transformations are done, we will again update our stored data at the backend with the frontend data of the user's choice.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Our project's main idea is ETL, i.e., extract, transform, and load. ETL comes into play when we have data that is incoming from various sources. These sources may not produce homogenous data. It may be required to transform such data into a standard format and then integrate it. The output of the project is a multi-dashboard data analysis and visualization platform. Multiple steps required for achieving the desired output are given below.

A. Extract

Given the modern rate and methods of data generation, the data formats are eclectic. There is input provision for all formats of data. The main data input sources are CSV, Excel, SQL databases, no-SQL databases, and API data. All of these sources have equal support and ease of execution. The user must be able to read and alter the data being loaded before actually loading the dataset. There is a preview provision. For extracting the data, we are using Python. Python has great connectivity with all the mentioned data sources. There are drivers or libraries available for the respective data sources. We are using Python through C# by using the python.NET package. This package creates a Python runtime engine, which we utilize through a Python scope. Then we run Python commands in the C# Python Runtime Engine and then convert this data to C# variables or objects.

B. Transform

After extracting the required data, the user can perform the desired actions on the data. The data is extracted in the form of Python Pandas' data frames. This way, no matter the source, the data can be transformed and integrated into a single structure. Transform operations such as dimension reduction, pivot, aggregation operations, merge, and map.

C. Load

The transformed data, at some point in time or form, might be crucial for a user. The user may need this transformed data for any purpose. So, the user can save this transformed data into a desirable format using the load operation. Reduce can be performed. The actions and their order are stored by the application. Transformed data from one source can be operated on with transform operations on transformed data from another source. There is no limit to this chain of transformations.

D. Script Generation

A dynamic script of the steps implemented by the user is generated to save the user's progress. The script can be re-run on re-login, and the user can continue their work from the left checkpoint. This script is generated in Python. The script generation is purely dynamic and will contain operations and code for the performed actions only, not all the available operations. This progress is shared with other users of the platform.

E. Dashboard Generation

All of the ETL operations have a GUI. All of the extracted data will be visible in the form of columns to the user. There is drag and drop feature available for data visualization. We drag the column to respective axes to get the graphs. There are various graphs which are available for use. The result from one graph can be used in another. There is also support for multiple dashboards.

F. API

API (application programming interface) is an important part of any software. It helps bridge communication between the front-end and the back-end. We've also implemented an API for the sake of data transfer between the user and the back-end (business logic). We've implemented various controllers for performing the tasks of carrying the data from server to client and vice versa. These controllers contain the logic of transforming the data based on which function is called, as well as the ability to fetch data sent by the client and send him an updated version of his or her data. Auth Controllers are responsible for login and signup; Script Controllers are responsible for Python script generation as well as going back to previous checkpoints, etc. This API is coded in the.NET Framework, or what we can call the ASP.NET Web API. The .NET framework is very flexible as it is an OOP language, which helps in implementing logic comparatively easily. Each controller has more than one API endpoint, which is mapped to various functions that are assigned different tasks. Once the client hits a particular endpoint, the call is shifted to the function that is mapped to that endpoint on the server side. Then the actual transformation of the data takes place. We've created generalized as well as specialized APIs that cater to general functionalities as well as any special transformations needed on the data as per clients' needs. This is why APIs play a major role in our project. Without the API, we wouldn't be able to transfer the data.

G. SQL

As we know, SQL is one of the most popular structured database services that most organizations are currently using. That is why we have added the functionality to connect to the client's SQL server to ease his need to perform analysis and visualizations on SQL data. If SQL as a data source was not provided, then the user would've had to export his database, convert it into a CSV file, and upload it. So, keeping in mind the end user's ease, we decided to have another option to upload the SQL data. We first take the user's SQL server name, other credentials, and the database in which he has the table on which he wishes to perform the analysis. Then, after setting up the connection with his or her SQL server, we display all the tables present in that particular database and give the user the choice to select the table of his or her choice. This way, we make it user-friendly and dynamic. Once he or she selects the table of his or her choice, we get this table at the backend, where we use Pandas to convert the SQL table into a data frame, which will eventually be used for further analysis. Once this is done, the user can now perform any of the operations on the data as well as visualize it with ease.

H. Python Script

The goal of this analysis dashboard is to save the checkpoints up to which the user has done the analysis. Instead of the traditional or currently in use techniques to save checkpoints, we're using a new method wherein, once the user has finished with his data transformation, we provide the user with a Python script that contains the code of all the transformations done till now.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

So, when the next time the same user visits the dashboard and wants to continue from the same checkpoint he left, he just needs to upload the Python script generated earlier, and he'll be jumped to the point where he last left off. This is one of the major differences between our analysis dashboard and the other existing conventional dashboards.

V. SYSTEM ARCHITECTURE

The system architecture mainly consists of a client end and a back end. The client end is where the user will provide the input data, i.e., through a CSV file, API, etc. Once the user enters the data, it will be generated at the back-end and stored in suitable data structures. The integration between the client end and the back end will be done through API calls implemented in C#. Again, at the client end, the user will perform operations as per his needs. All these operations are programmed in C#, into which Python is integrated using the Python.NET library. The end result of these operations will be transformed data that will be updated in the data structures. Then the transformed data will be transferred from the back end to the client end using an API call of the GET type.

Once the transformation of the data is done, the data will be presented to the client, who will be able to download it or share it wherever he wants. Also, along with the data, a Python script of the functions performed by the user, which is generated dynamically as the user performs them, is presented to the user. This will be done so that he can just input the Python script and perform the same tasks on different data automatically instead of manually performing them every time.

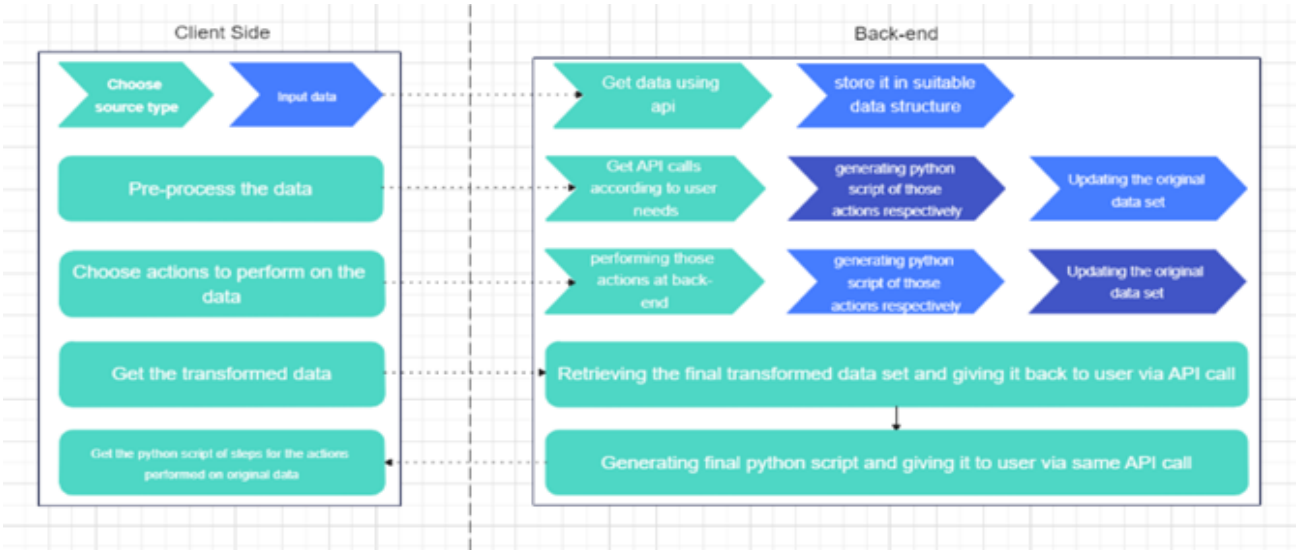

Fig. 1. System Architecture for Data Analysis Module

The above illustration displays the system architecture for the platform that has been developed. It shows the connection between the client and server sides.

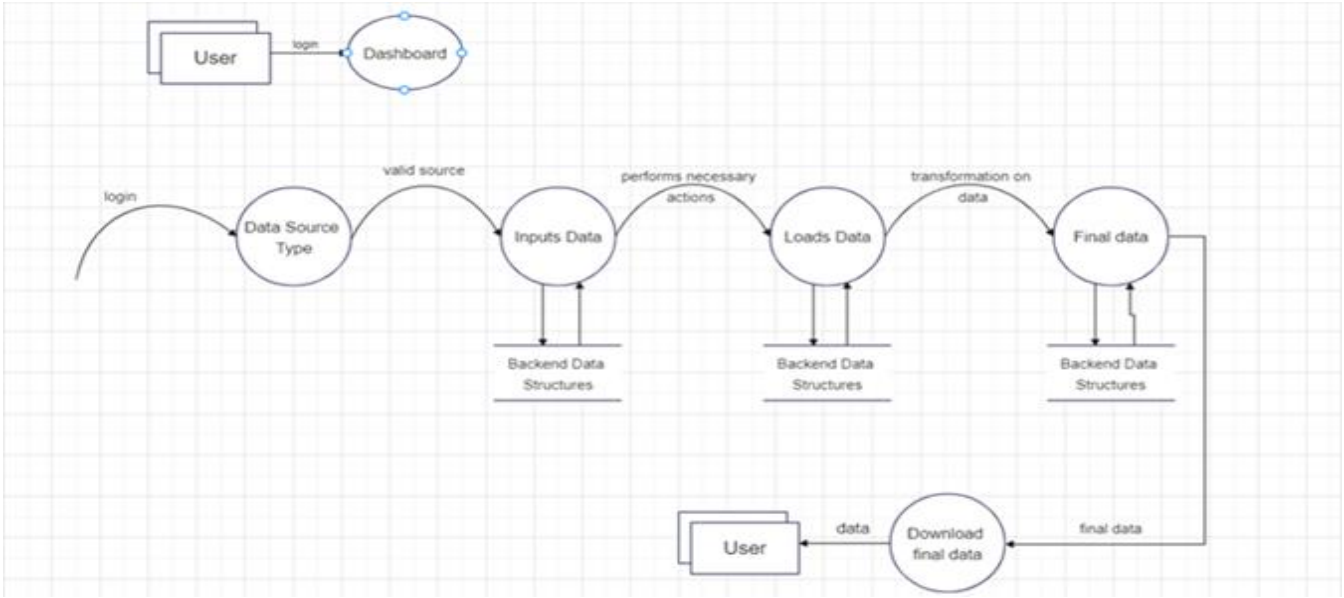

Fig. 2. Data Flow Diagram

The above illustration shows the flow of the data when the user uploads it and performs transformations on it.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

VI. ADVANTAGES & LIMITATIONS

There are various types of businesses out there that have various common needs that differ in deeper specifics. In a similar way, a data analysis dashboard must be not only up to par but also adaptable to any kind of data. An adaptable dashboard lets users perform tasks as per their needs and becomes fully business-centric. Since our dashboard is going to be web-based, or, you can say, deployed on a cloud as a SaaS, users do not need to download any apps or use high computing resources. This lowers the downtime and is easily deployed as well. On similar lines, we talked about users not needing high computing resources, but since it is cloud-deployed, the backend also does not require high computing power servers or data centres, which saves a lot of cost. Again, as the data involves mainly businesses, the clients might be busy throughout their day and may not have laptops or PCs with them all the time. The web-based and mobile-friendly UI facilitates their need to use the dashboard from anywhere on earth with internet connectivity, thus saving their precious time. Imagine that if you have multiple reports or datasets, you would have to spend a lot of time and effort reviewing them to reach your final goal. Our dashboard allows you to see all the analysis results under one roof. There is no need for clients to turn to third-party analysis software to perform analysis on their data. Through this project, we are providing an in-house solution for all users' needs. Users are able to get a decision-making model out of the data they give as input, making the data useful for future decisions. Since it is web-based and performs analysis on the given data at the current scenario or time, it doesn't update itself in real time if the given data changes. This means it is static in nature. So, this limitation can be overcome in the future by adding functionality that works on real-time data, similar to Google Sheets, which, when updated, reflect the changes in real time. It can be relatively less useful in this upcoming world of technology. The tech stack that we have used in the project is pretty different from current dashboards. Hence, it might be hard to understand when studying it for the first time. For a user from a non-technical background, he or she might not be able to understand how to use the features correctly and effectively, making the process time-consuming.

VII. RESULT AND DISCUSSION

The main goal of the application is to create a business intelligence and analysis platform. The application will have an efficient data flow pipeline. The data flow starts with the user entering the data, transforming the data, and loading the data according to the user's needs. The data won't be saved on the backend, so it is completely safe with the user. This will greatly help the organization, as it won't have to upload sensitive data anywhere and won't have to worry about a third-party app breaching their privacy. Considering the new technology stacks and emerging web services and technologies, there will be a significant improvement in the execution time and quality of the result. The application will be able to take input data and create interactive dashboards and graphs. This will give the data a visual representation. This visual representation will be able to give a better understanding of the trends, strengths, weaknesses, changes to be made, future growth, etc. These dashboards are shareable and have authorized access and access provisions. This platform will be accessible from multiple devices, so it will be operable from any device with an internet connection. This platform will be deployed in the cloud, so there won't be any requirement for high-power computational servers. This will greatly boost the productivity of the organization. There will also be a machine learning model that can be added in the future for future prediction, i.e., a decision recommendation based on the input data. With this, the organization will be able to adjust their goals, optimize their strengths, and convert their weaknesses into strengths. This will also impact the management sector of the organization, as there will be more effective communication of milestones, future expectations, and loss recovery if encountered.

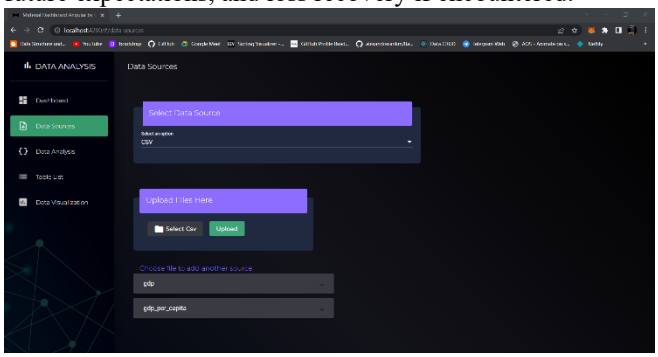

Fig. 3. Upload Files Page

This is the upload files page where the user can select the data source of their choice to be uploaded.

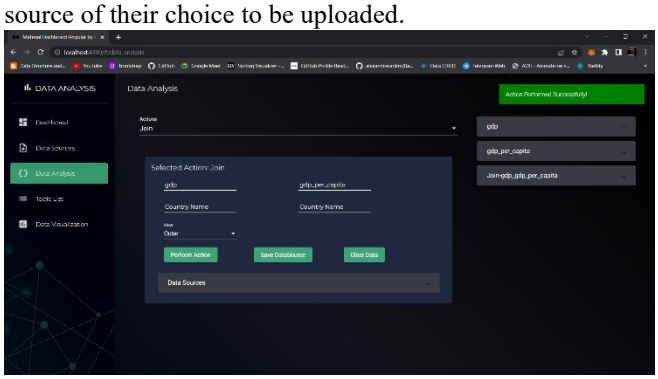

Fig. 4. Data Analysis Page

This is the data analysis page where the user will perform the actual transformations on the data.

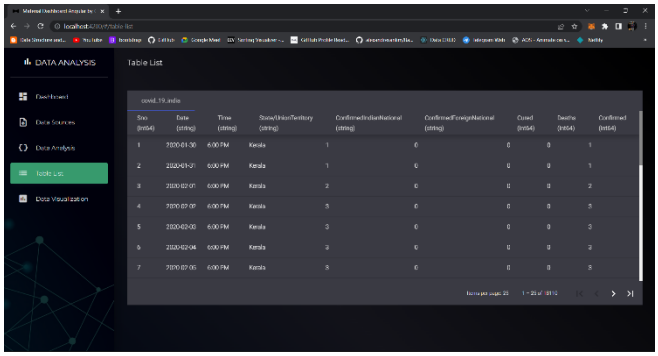

Fig. 5. Table List Page

This is the table list page where the user can see the data sources he uploaded in tabular form.

Retrieval Number: 100.1/ijitee.G96140612723
DOI: 10.35940/ijitee.G9614.0612723
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

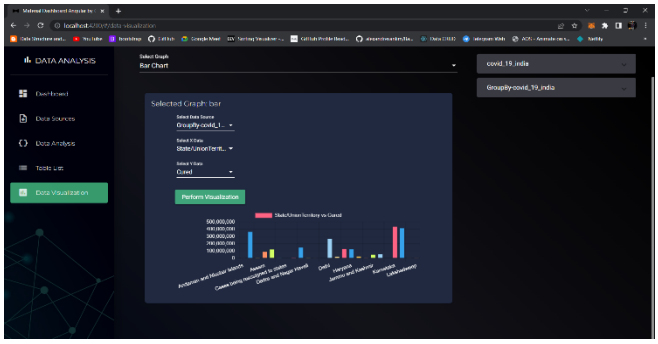

Fig. 6. Data Visualization Page

This is the data visualization page where the user can perform different kinds of visualization on his data to get inferences from the transformed data.

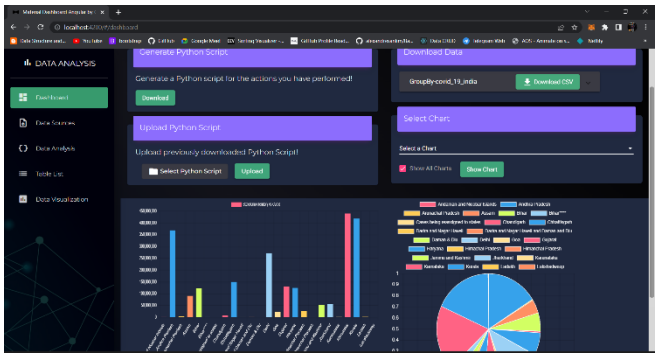

Fig. 7. Dashboard

This is the dashboard where the user can download the transformed data in the form of a csv, compare the visualizations performed, and get the Python script for all actions performed.

VIII. CONCLUSION

The module focuses on catering to the needs of all its users. The results provided by the module will help businesses make precise decisions. With features like visualization, one can understand trends in the markets as well. The software will provide its customers with all necessary tools for data analytics in an integrated dashboard. In this data-driven world, using data and producing useful insights is what this module will do. In the current scenario, only a limited number of operatives can actually work with these kinds of tools, but in the near future, customers from any background will be able to use the dashboard or software without any hassle. Integrating the module with pre-trained machine learning models will allow customers to dive deep into the analytics and get even more precise results.

DECLARATION

Funding/ Grants/ Financial Support	No, we did not receive.
Conflicts of Interest/ Competing Interests	No conflicts of interest to the best of our knowledge.
Ethical Approval and Consent to Participate	No, the article does not require ethical approval and consent to participate with evidence.
Availability of Data and Material/ Data Access Statement	Not relevant.
Authors Contributions	All authors have equal participation in this article.

REFERENCES

Journal reference

1. Dr. S. Ramakrishna, S. Sajida, "A Study of Extract–Transform– Load (ETL) Processes", 2015, International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181
2. Swacha, Jakub & Muszynska, Karolina. Python and C\#: A comparative analysis from Students' perspective, Annales umcs informatica lublin polonia, (2011), 11. 89-101. 10.2478/v10065-011-0023-6. [CrossRef]

Conference reference

3. V. C. Emeakaroha, N. Cafferkey, P. Healy and J. P. Morrison, "A Cloud-Based IoT Data Gathering and Processing Platform," 2015 3rd International Conference on Future Internet of Things and Cloud, 2015, pp. 50-57, doi: 10.1109/FiCloud.2015.53 [CrossRef]
4. N. Fotiou, V. A. Siris, A. Mertzianis and G. C. Polyzos, "Smart IoT Data Collection," 2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM). [CrossRef]

For website reference

5. Susnjak, T., Ramaswami, G.S. & Mathrani, A Learning analytics dashboard: a tool for providing actionable insights to learners. International Journal of Educational Technology in Higher Education 19, 12 (2022). https://doi.org/10.1186/s41239-021-00313-7 (2002) The IEEE website. [Online]. http://www.ieee.org/ [CrossRef]

AUTHORS PROFILE

Swarupa Deshpande pursued her M.E. (Computer Engineering) from Savitribai Phule Pune University, Pune in March 2017 with 7.96 CGPA. (First Class with Distinction), B.E. (CSE) from Shivaji University, Kolhapur in 2001 with 69.71% (First Class with Distinction). Currently working as Assistant Professor at Marathwada Mitra Mandal's College of Engineering Pune. Published research paper entitled "Fast Automated Detection of COVID-19 from CT Images Using Transfer Learning Approach" in Scopus indexed 1st International Conference on Intelligent Systems and Applications (ICISA 2022).

Atharva Gupte is pursuing a B.E. degree in Computer engineering from Savitribai Phule Pune University. Born in Pune on December 2000. He has a keen interest in Entrepreneurship and Management. Loves to trek and travel. Will be pursuing a Master's degree in Management in 2023

Kaushal Bhide is pursuing his B.E. degree in Computer engineering from Savitribai Phule Pune University. Born in Akurdi on 24th Feb 2002. He has a keen interest in web development. Also have done an internship as a web developer

Manas Apte is pursuing a B.E. degree in Computer engineering from Savitribai Phule Pune University. Born in Thane, Mumbai 2001. He keeps an interest in Data Science and is an aspiring Data Scientist.

Rohan Rasane is pursuing his B.E. degree in Computer engineering from Savitribai Phule Pune University. Born in Pune on 7th May 2001. He has a keen interest in programming and AI.