iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.81413

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Design and Implementation of Dataverse: A Web-Based Dataset Management Platform for ML/DL

N. Sai Teja[1], M. Amulya[2], K. Jagadeesh, B. Jagadeesh[3], B. Gopal[4]
[1, 3, 4]*Department of Artificial Intelligence and Machine Learning Final year Major project -- 2026 , Acharya Nagarjuna University, Guntur, Andhra Pradesh, INDIA*
[2]*M.Tech., Assistant Professor, Department of CSE*

Abstract: ***Data processing and visualization are crucial steps in data processing that greatly affect the quality of future analyses and machine learning algorithms application. Still, such data processing steps may be tedious and labour-intensive, and often require human effort which is associated with a lot of mistakes when users do not have sufficient computer science knowledge.***
The current research presents the development of a web based system called Dataverse. The purpose of developing this system is to automate and facilitate the process of data preparation and provide an opportunity for the user to explore the prepared data through interactive visualization.
Dataverse is developed using Python programming language in Flask framework. The libraries employed during Dataverse development are mainly Pandas and Plotly. The system provides opportunities for importing datasets from files and online resources.
According to experiments conducted, Dataverse facilitates the process of preparing data and makes it less labour-intensive and more efficient.
Keywords: Data Preprocessing, Data Preparation, Flask, Data Visualization, Web Applications, Interactive Analytics, Datasets

I. INTRODUCTION

In the modern age of technology, data forms the bedrock of all software, which includes business analytics and artificial intelligence. The efficiency of the algorithms, models, and predictions to some extent depend upon the dataset used in the process. However, the data available in reality might not always be complete and consistent, hence leading to the necessity for data preprocessing. Manual preprocessing could prove to be very time-consuming and prone to errors; even experienced analysts sometimes face difficulty in working without programming skills.

The present project suggests a web application called Dataverse, which is created using the Flask framework and will help to automate the process of data cleaning along with interactive data visualization. The user will input their data and then the application will detect any issues (like missing entries and duplicate rows) within the data and offer visualization to proceed further.

The software enables the process of making pre-analysis simpler since all the required activities can be performed automatically; at the same time, it allows the user to receive immediate feedback through the Internet browser.

It acts as a link between unprocessed data and information that can be analysed further.

II. LITERATURE REVIEW

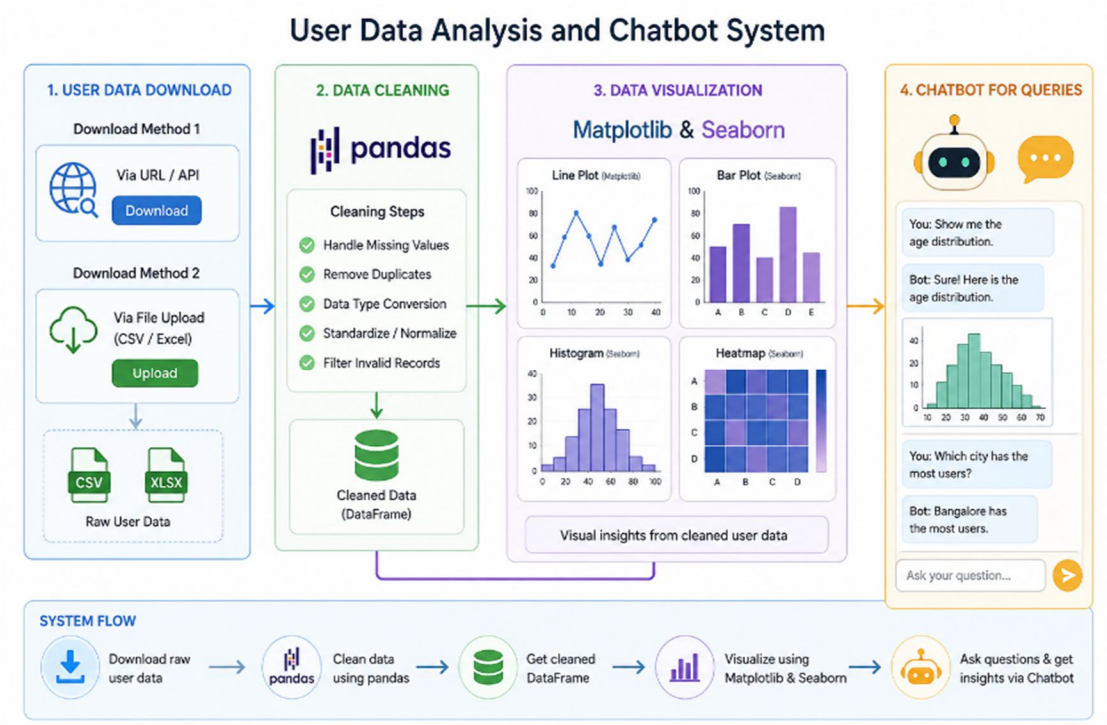
There are numerous tools that can be utilized for data cleansing and visualization purposes, where each tool can solve specific problems; however, there are no tools that can make the entire process easier for users. Tools such as Open Refine and Tableau Prep can give users the opportunity to work with any inconsistencies in the data visually.

However, these two tools are difficult to integrate and apply or are licensed software solutions. Some tools, including Auto Clean and Wrangler, aim to automate the process of data pre-processing. However, the mentioned tools cannot give users the option to Visualization in an integrated way. Moreover, there is the problem with using command-line or a desktop application for handling the task. Even though the emergence of web programming frameworks such as Flask and Django provided means for creating flexible and open-ended tools for solving the issue, most of these solutions only tackle each part of the process separately.

To overcome these limitations, *Dataverse* is designed as an integrated web-based platform that brings automated preprocessing and interactive visualization together in a single environment. By combining these functionalities, the system offers a more accessible and adaptable solution for students, researchers, and analysts, helping them efficiently transition from unprocessed data to meaningful insights.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

III. SYSTEM DESIGN



The Dataverse concept can be described as a modular web application in which users can upload their datasets, apply automatic preprocessing on them, and generate visual analysis through the web browser. This application follows a client-server architecture that offers flexibility and easy implementation. The server-side is built using Python and the Flask framework.

A. Brief Overview

The system is organized into three primary components:

1) User Interface (Frontend): It consists of a web-based dashboard through which users can upload the datasets and also view the processed output in an intuitive way

2) Application Backend (Flask): Responsible for managing requests, executing preprocessing operations, and coordinating communication between components.

3) Data Processing & Visualization Module: It takes care of data cleaning and visualization.

B. Workflow

The process starts after the submission of the file by the user in various formats, such as CSV or XLSX. The system will verify the file and conduct the preprocessing process, which involves handling missing data, eliminating duplicates, and formatting the data. Once the preprocessing process ends, the system generates the output, which includes the cleansed data set and the visualization of the data. The generated output will be presented through the web interface that users can comprehend without having programming knowledge.

C. Data Flow Description

There are many different stages or processes in data flow that primarily aim at increasing efficiency in data management or utilization by the user. The first step involves submission of the data set by the user through the web portal interface. This means that the user can submit his/her data either by uploading the data or using a valid link to access the data source.

Once the submission is done, the data set will be temporarily stored in the server and undergoes several tests to verify whether the data set is accurate and suitable for use. Preprocessing is another activity that takes place whereby formatting is done among others.

D. Tools And Technologies Used

Module	Technology Used
Frontend	HTML,CSS,JavaScript
Backend	Flask(Python)
Data Cleaning	Pandas, NumPy
Visualization	Matplotlib
Storage	Local storage

E. Advantages

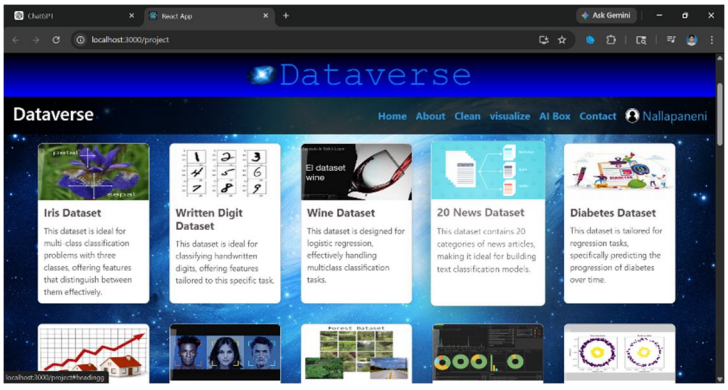
In this way, it was developed so that it will be easy for people to operate within it even when they have never dealt with software for manipulating data before in their lives and will be able to do all sorts of complicated things in there without needing to know anything about programming and algorithms.

The other reason why the tool is quite useful is that it can deal with certain basic preprocessing tasks that will save a person a lot of time and effort as well as prevent him/her from making unnecessary mistakes while performing those operations manually.

The other valuable feature of the software tool is that one will not need to develop scripts for working with the information since everything will be automated.

Lastly, the system was built in such a way that it will be possible to upgrade it with some additional features in the future.

IV. IMPLEMENTATION



The reason for choosing to use Flask, a web framework of Python language, in the construction of Dataverse lies in the lightness of the tool and its flexibility when developing web applications. The adoption of Flask allows one to build a responsive backend which will react to user's needs and perform necessary actions on datasets, and help interact with all elements. Flask is characterized by its lightweight nature, giving developers full control over it regardless of scalability.

In general, thanks to Flask and the efficiency of Python in processing datasets, Dataverse provides an efficient service. The latter implies dataset management starting from their upload until the moment they become easily visualizable.

A. Backend Development

Backend of Dataverse serves as the core component of the project where all the processes of data processing occur together with communication between front-end and different components of the system. Backend development was performed in the app.py file using the Flask framework where different routes were defined for such operations as loading datasets, preprocessing, and visualization of the results.

The first step of the process begins with the user uploading a dataset or linking it directly to the application. Backend loads the data in the Pandas DataFrame format and performs the necessary validation of the uploaded information. Following this, the process of preprocessing starts involving dealing with missing values, elimination of duplicates, and normalization of columns. Additional processing is performed according to each particular data type.

Once preprocessing is successfully completed, the processed dataset is then organized in a structured format (for example, JSON).

B. Data Preprocessing Logic

An excellent preprocessing technique has been developed which will prepare the data set for visualization as well as machine learning purposes. A few of the key steps involved in preprocessing the data are as follows:

1) Missing Value Imputation: Automatic imputation of missing numeric data values using their average and missing categorical data values using blank space.
2) Removal of Duplicate Data Rows: This step ensures consistency in data and reduces redundant entries.
3) Header Consistency Check: Consistency check of header datatypes.
4) Image Data Identification: Identification of columns containing image data through filename and URLs.

The steps ensure that the data is standardized. Dataset..

C. Frontend Functionality

The front end code used in Dataverse is developed in React technology. This helps in developing a dynamic interface because of the component oriented nature of React code. There are different components that can help manage different sections of the interface, hence making it easy to manage the interface because of increased access to the program.

In the main interface, there is display of both the dataset and its processed form. Users will also get information about the dataset such as the number of rows and column present within the dataset, even if there is missing data. Information obtained from this helps them gain some insight into the data set.

Communication between the front end and the backend is achieved using API calls. Simply put, this entails the process of sending data from the React frontend to the Flask backend which processes the same before sending back the results to the user interface. Changes will be dynamic.

D. Workflow Summary

All stages of the Dataverse processing cycle can be split into a number of steps that allows the code to remain clear, maintainable, and simple:

1) Input Stage: Everything begins with the user uploading the necessary dataset either in the form of a file upload or a valid link to the dataset through the frontend. If the input is in a compatible format and has passed validation successfully, it will be further processed.
2) Processing Stage: As soon as the input stage has been completed, the backend runs its logic with the help of Flask in the app.py file that is responsible for handling requests.
3) Transformation Stage: With the help of the Pandas library, the input dataset undergoes transformation, including data preprocessing and validation.
4) Output Stage: At last, the processed dataset is returned to the frontend in order to be displayed to the user.
5) Generally, such an architecture is maintainable and scalable enough for further additions of new features, including more advanced visualization tools.

E. Testing

The program was tested on different combinations of the input data. This data comprised of CSV files and images, which allowed evaluating the appropriateness of the tool on different kinds of input files. The testing was carried out according to different parameters, such as the size of the dataset and the ratio of the missing data.

All major elements of the program worked properly and showed excellent results, such as flawless dataset loading, preprocessing of the files and viewing the results of processing. Also, one of the most significant advantages is the high speed of the process when working with moderately large databases.

V. RESULTS AND DISCUSSION

This program offers a unified approach to data preprocessing and visualization in one internet program. In many other cases when the processes are carried out separately, using this software is considerably more convenient as the two activities take place at the same time. While conducting testing of the program, several sets of structured data, numeric and pictorial, were analyzed.

Tests reveal that this program is always reliable and accurate in any case. It is able to provide precise results of data pre-processing along with rather quick processing time irrespective of the size of the set used. Taking everything into account, it is safe to state that the Dataverse is an efficient tool that is worth recommending.

A. *Experimental Results*

It can be seen that Dataverse is capable of identifying and solving common issues related to the dataset in an automated manner. The inputted datasets will be analysed quickly and will display the cleansed datasets in the same interface; therefore, it will not require any assistance at all.

The processings involved in the process include the following steps:

1) Deleting duplicate rows so that the data will remain unique.
2) Changing image URL with respect to their thumbnails.
3) Properly naming and formatting columns so that they remain consistent throughout.
4) Exporting the cleansed datasets into other formats for reuse.

There is also a main interface where the raw and processed datasets will be displayed in a tabular form. The software performance analysis results in the processing time taking less than five seconds, provided that the size of the datasets falls under the range of 10,000 to 50,000 rows.

B. *Discussion*

In addition, these findings suggest that numerous data preprocessing steps can be easily automated through the use of data processing functions made possible by Python programming libraries in combination with a web application framework like Flask. The modularity of the system implies that adding new features, such as outlier detection, encoding features, and enhanced visualization of data sets, could be done with ease during future upgrades.

Currently, Dataverse supports both numerical and textual data sets and therefore can be used to analyze data available in different formats. Nevertheless, due to its versatile nature, it is highly suitable for the integration of complex elements of artificial intelligence and machine learning systems that would enable more complicated data analysis tasks to be performed.

Given that Dataverse provides a no-code data management system, it makes sense why it is ideal for everyone who needs to manage data regardless of their professional background, whether researchers, data scientists, or machine learning experts.data.

VI. CONCLUSION AND FUTURE WORK

Indeed, this project succeeded in developing and implementing a Dataverse dataset preprocessing and visualization software. It is worth noting that the backend framework used to develop this software included Flask, while Python packages such as Pandas and NumPy were used to undertake various data preprocessing activities effectively.

One of the distinguishing characteristics of the proposed software is that it is very easy to use because it enables users to upload, preprocess, and view a preview of the datasets. With the help of the software application, users can perform various preprocessing tasks automatically, including dealing with missing values, avoiding duplication, standardizing, and viewing the preview of images in datasets. Notably, one of the great advantages of the system under review is that it promises to have a high level of usage in academic institutions and scientific research owing to its ease of use and absence of coding. In the future, additional features can make it an even more powerful system.

VII. FUTURE IMPROVEMENTS

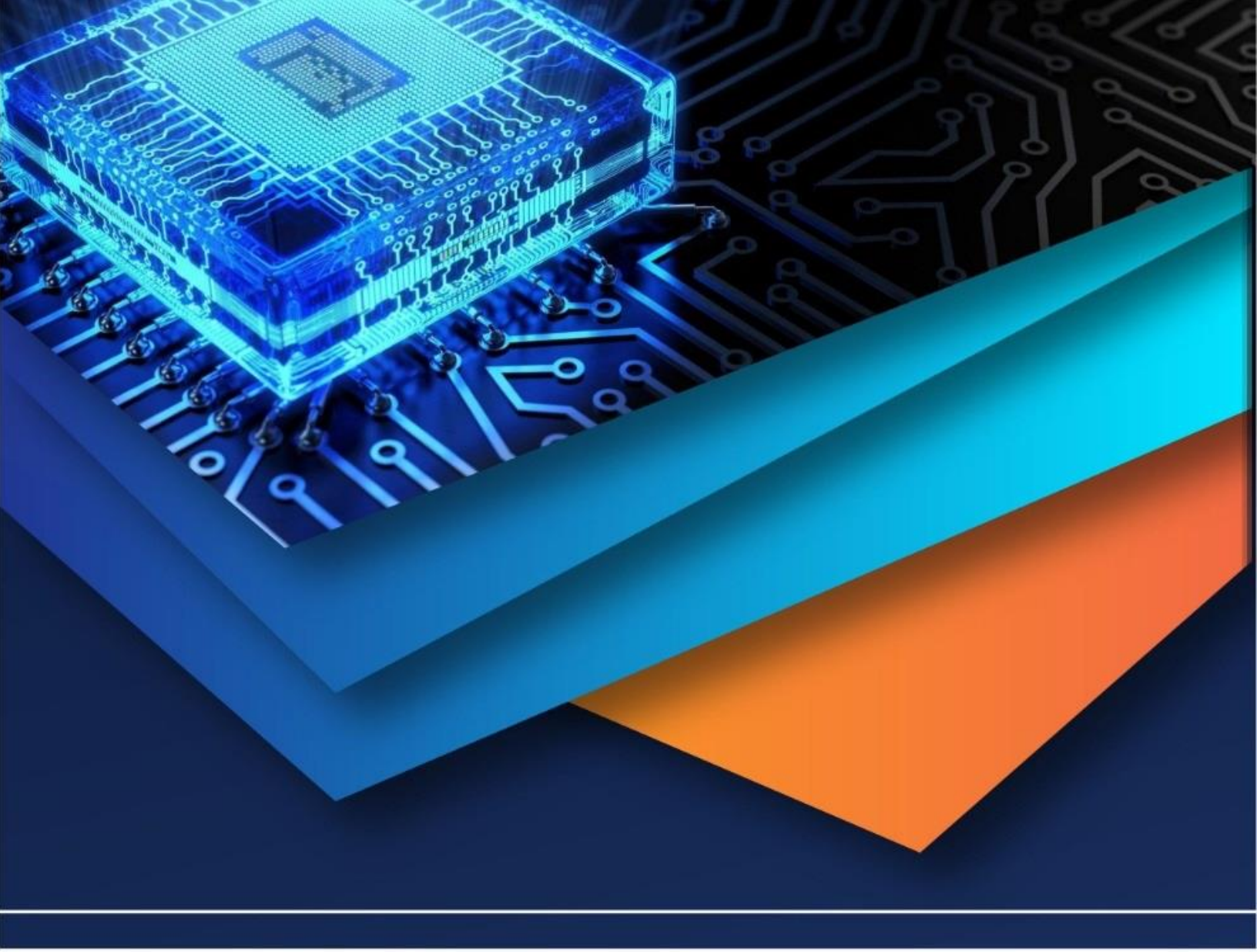
However, while the system is quite efficient at achieving its intended functions, it may benefit from a number of modifications in future versions. Firstly, one area in which improvements should be made is incorporating visualization abilities via the creation of visual tools such as graphs and plots using software packages like Plotly and Chart.js.

Moreover, another extension to the project could be the incorporation of machine learning abilities in order to predict trends within the dataset. An additional area for expansion of the current system is the ability to authenticate users as well as allow user profiles in order to track dataset history

REFERENCES

[1] S. Kandel, A. Paepcke, J. Hellerstein and J. Heer,“Wrangler: Interactive Visual Specification of Data Transformation Scripts,” Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI), pp. 3363–3372, 2011. Vis Stanford
[2] OpenRefine Project, “OpenRefine— a free, open source tool for working with messy data,” OpenRefine. [Online]. Available: https://openrefine.org/. OpenRefine
[3] “Cleaning Data with OpenRefine,” Programming Historian. [Online]. Available: https://programminghistorian.org/en/lessons/cleaning-datawith-openrefine. Programming Historian
[4] Pallets Projects, “Welcome to Flask — Flask Documentation.” [Online]. Available: https://flask.palletsprojects.com/. Flask Documentation+1

[5] The pandas development team, “pandas: Python Data Analysis Library— Documentation,” pandas, 2025. [Online]. Available: https://pandas.pydata.org/docs/. Pandas+1
[6] NumPy Developers, “NumPy Documentation,”NumPy. [Online]. Available: https://numpy.org/doc/. NumPy+1
[7] Plotly Technologies Inc., “Plotly.py— Python graphing library,” Plotly Documentation. [Online]. Available: https://plotly.com/python/. Plotly+1
[8] Zenodo, “Zenodo— preserve and share research outputs (DOI & GitHub integration),” Zenodo. [Online]. Available https://zenodo.org/. Zenodo+1
[9] D. B. Cleveland, “Data Visualization: Principles and Practice,” Hobart Press, 1993.
[10] M. Friendly,“ A Brief History of Data Visualization,” in Handbook of Data Visualization, Springer, 2008, pp. 15–56.
[11] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY