iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** V **Month of publication:** May 2023

DOI: https://doi.org/10.22214/ijraset.2023.52211

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 11 Issue V May 2023- Available at www.ijraset.com

Image Metadata Viewer or Editor

Aldo Tenis A[1], Johans Praveen S[2], Karunya MD[3], Preethiga M[4], Vinoth Kumar A[5]
Computer Science and Engineering, Ssm Institute Of Engineering And Technology (Anna University)

Abstract: ***Our project aims to provide a solution for viewing and editing metadata, which is the concealed data embedded within image files. Metadata typically includes crucial information such as the date and location where the photo was taken, the author's details, camera specifications, and other relevant data. By leveraging our project, users can easily access and modify this information. This enhances the confidentiality of the image and allows for safe sharing of images on social media platforms without any concerns. Many users are unaware of the importance of metadata in image files, which can contain critical information such as the location, date, and time of the photo's capture, as well as camera settings and other relevant data. This information can be essential for organizing and categorizing images, as well as for legal and security purposes. However, most users lack the tools and knowledge required to view and edit this metadata, which can lead to privacy and security concerns when sharing images online. Therefore, is the need for a user-friendly and efficient solution that allows users to access and edit image metadata easily, enhancing the confidentiality and security of their images while enabling safer sharing on social media platforms.***

Keywords: Metadata, Confidentiality, File Security, User Friendly, Social Media

I. INTRODUCTION

Metadata refers to the descriptive information about data, such as the date it was created, the author, location, and format. In today's digital age, metadata has become an integral part of data management, allowing organizations to efficiently organize, search, and retrieve data. However, metadata also raises privacy concerns, as it can reveal sensitive information about individuals and organizations. One of the primary privacy issues with metadata is that it can be used to track an individual's online activities. For instance, metadata embedded in emails can reveal the sender's location, device information, and online activity. Similarly, metadata in social media posts can reveal personal information, such as an individual's interests, relationships, and location.

A. Existing System – Metadata2go

It is a tool for viewing metadata of image files. The site provides a simple and user-friendly interface that allows users to upload their image files, and then it displays the metadata associated with the image. The metadata that can be viewed using this tool may include information such as the camera make and model, the date and time the photo was taken, the exposure time, focal length, and other technical details of the image. It may also display other relevant information such as the location where the photo was taken, the author of the photo, and other descriptive data.

Drawbacks

1) *Limited Functionality:* The website appears to only offer a basic service of viewing metadata associated with image files. It does not seem to offer any editing or manipulation capabilities, which may be a limitation for users who need to modify the metadata of their images.
2) *File Size Restrictions:* The website may have restrictions on the size of the image files that can be uploaded for metadata viewing. This may limit the ability of users to view metadata for larger files or batches of files.
3) *Limited Format Support:* The website may not support all image file formats, which may be a limitation for users who work with less common file types.

B. Proposed System

Our project aims to provide a solution for viewing and editing metadata in image files. This solution will be user-friendly and efficient, allowing users to easily access and modify metadata, which will enhance the confidentiality and security of their images. By providing a user-friendly tool for editing and viewing metadata, we aim to improve users' knowledge and awareness of the importance of metadata in image files.

©IJRASET: All Rights are Reserved

This will enable safer sharing of images on social media platforms and will also help users to organize and categorize their images more effectively. Our solution will leverage advanced technologies to provide a seamless and easy-to-use interface, ensuring that users can easily access and edit metadata without any hassle.

Advantages

1) *Increased file Format Support:* Expanding the range of file formats would make the tool more versatile and useful for a wider range of users
2) *Enhanced Privacy and Security Measures:* To address privacy concerns, we could implement additional security measures such as encryption or two-factor authentication.
3) *Improved Editing Capabilities:* Adding more advanced editing features, such as the ability to edit the metadata and modify the timestamps of the file will increase the confidentiality
4) *User Interface Improvements:* Improving the user interface could make the tool more intuitive and easier to use, which would improve the overall user experience

II. LITERATURE SURVEY

A. Forensic Analysis of Video Files Using Metadata

AUTHORS: Ziyue Xiang; János Horváth; Sriram Baireddy; Paolo Bestagini; Stefano Tubaro; Edward J. Delp

YEAR: September 2021The unprecedented ease and ability to manipulate video content has led to a rapid spread of manipulated media. The availability of video editing tools greatly increased in recent years, allowing one to easily generate photo-realistic alterations. Such manipulations can leave traces in the metadata embedded in video files. This metadata information can be used to determine video manipulations, brand of video recording device, the type of video editing tool, and other important evidence. In this paper, we focus on the metadata contained in the popular MP4 video wrapper/container. We describe our method for metadata extractor that uses the MP4's tree structure. Our approach for analyzing the video metadata produces a more compact representation. We will describe how we construct features from the metadata and then use dimensionality reduction and nearest neighbor classification for forensic analysis of a video file. Our approach allows one to visually inspect the distribution of metadata features and make decisions. The experimental results confirm that the performance of our approach surpasses other methods.

B. OME: Tool for generating and managing metadata to handle BigData

AUTHORS: Ranjeet Devarakonda; Biva Shrestha; Giriprakash Palanisamy; Les Hook; Terri Killeffer; Misha Krassovski; Tom Boden; Robert Cook

YEAR: January 2015

The next-generation On-line Metadata Editor (OME) is an easy-to-use tool to help document scientific data in a well-structured popular metadata format. In this paper, we discuss the newest tool that Oak Ridge National Laboratory has developed to input, edit, and manage metadata and how it is helping data intensive science centers across many federal agencies to prepare metadata and to make their BigData discoverable. OME is based on a metadata model that uses ontologies to represent metadata about data and data-related artifacts. The tool includes a user interface for creating and managing metadata, as well as a set of APIs for integrating with other data management systems. The paper describes several use cases for OME, including managing metadata for large scientific datasets, managing metadata for media assets, and managing metadata for a large-scale clinical research study. In each case, the authors demonstrate how OME can be used to manage the metadata associated with the data, and how it can facilitate data discovery, sharing, and reuse. The paper concludes by highlighting the benefits of using OME for managing metadata in the context of big data. These include improved data discovery and reuse, increased efficiency in managing metadata at scale, and better support for data management workflows. The authors suggest that OME has the potential to be a valuable tool for organizations that are dealing with large and complex data sets.

C. Metadata Editing and Mapping within the AXMEDIS Framework

AUTHORS: Kia Ng; Minh Thang Dang; Tran Vu Pham; Royce Neagle; Bee Ong

YEAR: December 2007

The paper "Metadata Editing and Mapping within the AXMEDIS Framework" describes the implementation of a metadata editor and mapping tool within the AXMEDIS (Automated eXchange of Media and Metadata in e-Services) framework. The paper was published in the Journal of Universal Computer Science in 2008.

The authors begin by discussing the importance of metadata in digital content management, and the challenges associated with creating and managing metadata in a distributed and heterogeneous environment. They note that the AXMEDIS framework was designed to address these challenges by providing a set of tools and services for managing multimedia content and metadata in a distributed and interoperable way.

The paper focuses on the implementation of two key components of the AXMEDIS framework: the AXMEDIS metadata editor and the AXMEDIS metadata mapper. The metadata editor is a web-based tool that allows users to create, edit, and validate metadata for multimedia content. The metadata mapper is a tool that allows users to map metadata between different metadata schemas and vocabularies.

The authors describe the design and implementation of these tools, including their user interfaces, features, and functionality. They also present a case study of the use of the AXMEDIS metadata editor and mapper in a real-world project involving the creation and management of metadata for a collection of multimedia content.

The paper concludes by highlighting the benefits of the AXMEDIS framework and its metadata editing and mapping tools for digital content management. These include improved interoperability, increased efficiency, and better support for multimedia content and metadata management workflows. The authors suggest that the AXMEDIS framework has the potential to become a valuable tool for organizations that are dealing with large and complex multimedia content collections.

D. Design and Implementation of the Metadata Modification Concept Minimizing File Modification

AUTHORS: YoungJun Yoo; Sun Sopharath; Yunhee Woo; Jin Kim; YoungWoong Ko

YEAR: February 2019

In recent years, the capacity of files such as documents and multimedia has been greatly increased. The user not only stores and reads data, but also performs various activities such as adding or deleting contents to an existing file. For editing, the file system copies all the contents of the disk into memory, and the user modifies this data using the application. Then, when storing the data again, the user application is using the write function to rewrite it all over the disk from beginning to end. In the vim editor, even when some data is modified, all data is saved again, resulting in slower performance as the file size increases. An example of another method is tar. tar consists of several sub-files, and if you delete one of these sub-files, all the sub-files after the deleted file are pulled forward and rewritten. This approach requires a lot of disk-write requests despite the small number of data changes, and it takes a long time to execute. In this paper, we proposed the Metadata Modification technique, which is a new concept of modifying part of a file quickly through some metadata modification. In this modification we will use a function call sys_meta_modification(). This function is a prototype, assuming a situation in which a portion of the file is deleted, and although it is a limited function, it treats the file in a way that is fundamentally different from the API of the original primitive file system. This function modifies the pointer in the metadata when the data is deleted on a block-by-block basis and returns the block to be deleted to the kernel. In other words, because there is no need to write back data to a disc from beginning to end, it was very quick to delete the contents of the file. It also shows that if you create a user application that imitates tar using this system call and deletes the sub-file, the data will be deleted quickly.

E. Decentralized and financial approach to effective charity

AUTHORS: Pratyush Agarwal, Shruti Jalan, Dr. Abhijit Mustafi

YEAR: February 2021

The paper "Use of a metadata documentation and search tool for large data volumes: The NGEE arctic example" describes the use of a metadata documentation and search tool for managing large volumes of scientific data in the context of the Next-Generation Ecosystem Experiments (NGEE) Arctic project. The paper was published in the Journal of Environmental Informatics in 2017.

The authors begin by discussing the importance of metadata in scientific data management, and the challenges associated with managing metadata for large and complex data sets. They note that existing metadata tools often require a significant amount of time and effort to use, which can limit their effectiveness for managing large data volumes.

The NGEE Arctic project involved the collection and management of large amounts of data related to the Arctic ecosystem. To manage this data, the project team developed a metadata documentation and search tool called the NGEE Arctic Metadata Explorer (NAME). NAME is a web-based tool that allows users to create, edit, and search metadata for scientific data sets.

The authors describe the design and implementation of NAME, including its user interface, features, and functionality. They also present a case study of the use of NAME in the NGEE Arctic project, highlighting how it was used to manage and search metadata for large data sets related to soil, vegetation, and permafrost.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

The paper concludes by highlighting the benefits of using a metadata documentation and search tool like NAME for managing large scientific data sets. These benefits include improved data discovery, increased efficiency in managing metadata, and better support for scientific data management workflows.
The authors suggest that NAME has the potential to become a valuable tool for other scientific data management projects dealing with large and complex data volumes.

III.REQUIREMENT AND ANALYSIS

A. Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. A feasibility study allows project managers to investigate the possible negative and positive outcomes of a project investing too much time and money. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

1) Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement as only minimal or null changes are required for implementing this system.

2) Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3) Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

4) Legal Feasibility

This study investigates if the proposed system conflicts with legal requirements like data protection acts or social media laws.

5) Scheduling Feasibility

In this feasibility study, we estimate how much time the system will take to complete, and with our technical skills we need to estimate the period to complete the project using various methods of estimation.

B. Hardware Requirements

1) *Processor:* Intel Core i5 or i7
2) *Memory (RAM):* At least 8GB of RAM is recommended may be necessary.
3) *Storage:* A Solid-State Drive (SSD) is recommended for faster read and write speeds.
4) *Display:* A high-resolution display is recommended to view and edit images accurately.
5) *Operating System:* The metadata editor can be developed on any modern operating system, such as Windows, MacOS, or Linux, depending on the developer's preference and target users.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

C. Software Requirements

1) Operating system : Linux
2) Front End : HTML, CSS
3) Back end : Flask
4) Database : Firebase

D. Functional Requirements

A functional requirement defines a function of a system or it components. A function is described as a set of inputs, the behaviour and outputs. Functional requirement deals with what the system should provide for users. They include description of the required functions, outlines of associated reports or online queries and details of data to be held in the system.

In our system the functional requirements are,

- ❖ Displaying the image/file uploading page
- ❖ Displaying the metadata of the uploaded file
- ❖ Displaying the edited file for download

Input : Image / File
Behaviour : Metadata can be hidden and users can download the new file.
Output : Displaying the edited file for download.

E. Non-Functional Requirements

The definition for a non-functional requirement is that it essentially specifies how the system should behave and that it is a constraint up on the systems behaviour.

Performance : Reliability, Accessibility, Confidentiality
Cost : Free

Non-functional requirements include the following also:

- The system shall be user friendly and consistent.
- The system shall allow developer access to installed environment.

F. System Architecture

An allocated arrangement of physical elements which provides the design solution for a consumer A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behavior) between them.

It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture; collectively these are called architecture description languages (ADLs).

Various organizations define systems architecture in different ways, including:

- Product or life-cycle process intended to satisfy the requirements of the functional architecture and the requirements baseline.
- Architecture comprises the most important, pervasive, top-level, strategic inventions, decisions, and their associated rationales about the overall structure (i.e., essential elements and their relationships) and associated characteristics and behaviour.
- If documented, it may include information such as a detailed inventory of current hardware, software and networking capabilities; a description of long-range plans and priorities for future purchases, and a plan for upgrading and/or replacing dated equipment and software

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

The composite of the design architectures for products and their life-cycle processes.

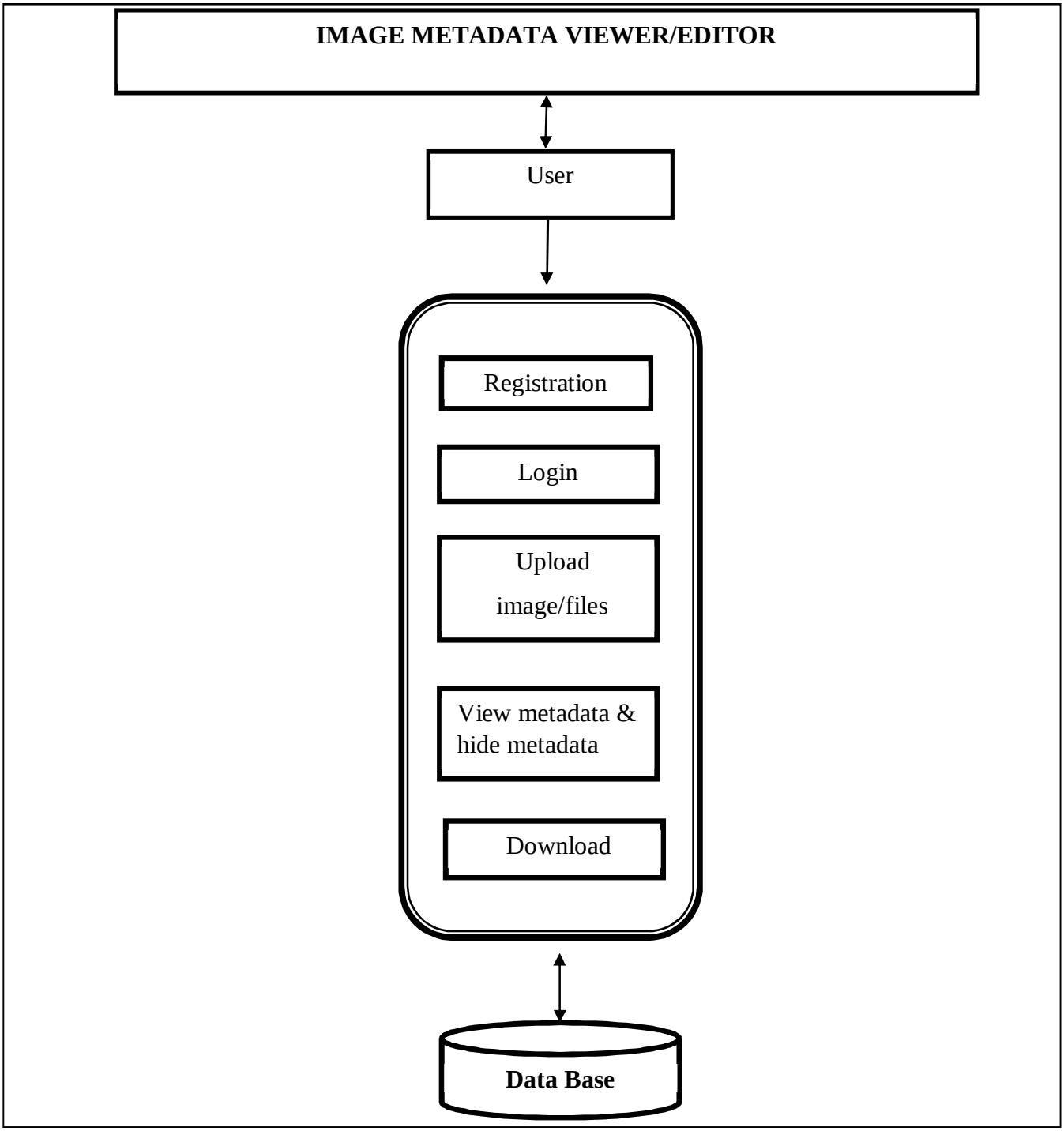
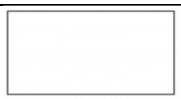


Fig 1. System Architecture

1) Data Flow Diagram

A data flow diagram is a two-dimensional diagram that explains how data is processed and transferred in a system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output. Individuals seeking to draft a data flow diagram must identify external inputs and outputs, determine how the inputs and outputs relate to each other, and explain with graphics how these connections relate and what they result in. This type of diagram helps business development and design teams visualize how data is processed and identify or improve certain aspects.

Data flow Symbols

Symbol	*Description*
	An **entity**. A source of data or a destination for data.
	A **process** or task that is performed by the system.
	A **data store**, a place where data is held between processes.
	A **data flow**.

- LEVEL 0

The Level 0 DFD shows how the system is divided into 'sub-systems' (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system.

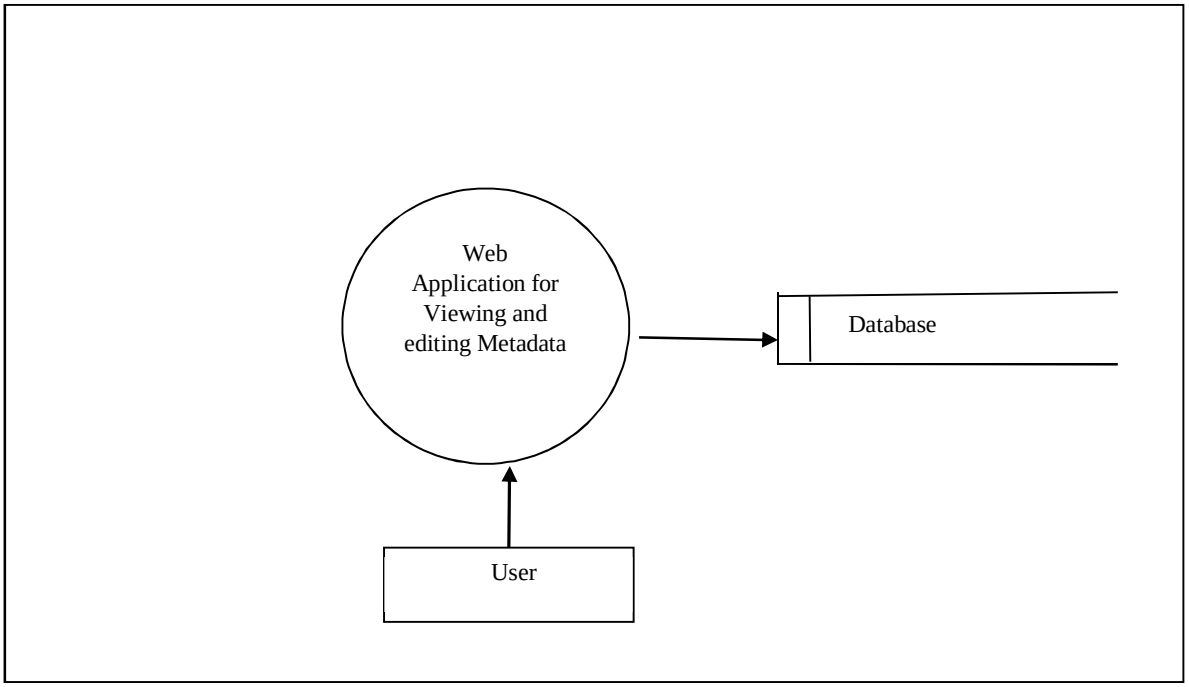


Fig 2. Connecting of Data

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

- *LEVEL 1*

The next stage is to create the Level 1 Data Flow Diagram. This highlights the main functions carried out by the system. As a rule, to describe the system was using between two and seven functions - two being a simple system and seven being a complicated system. This enables us to keep the model manageable on screen or paper.

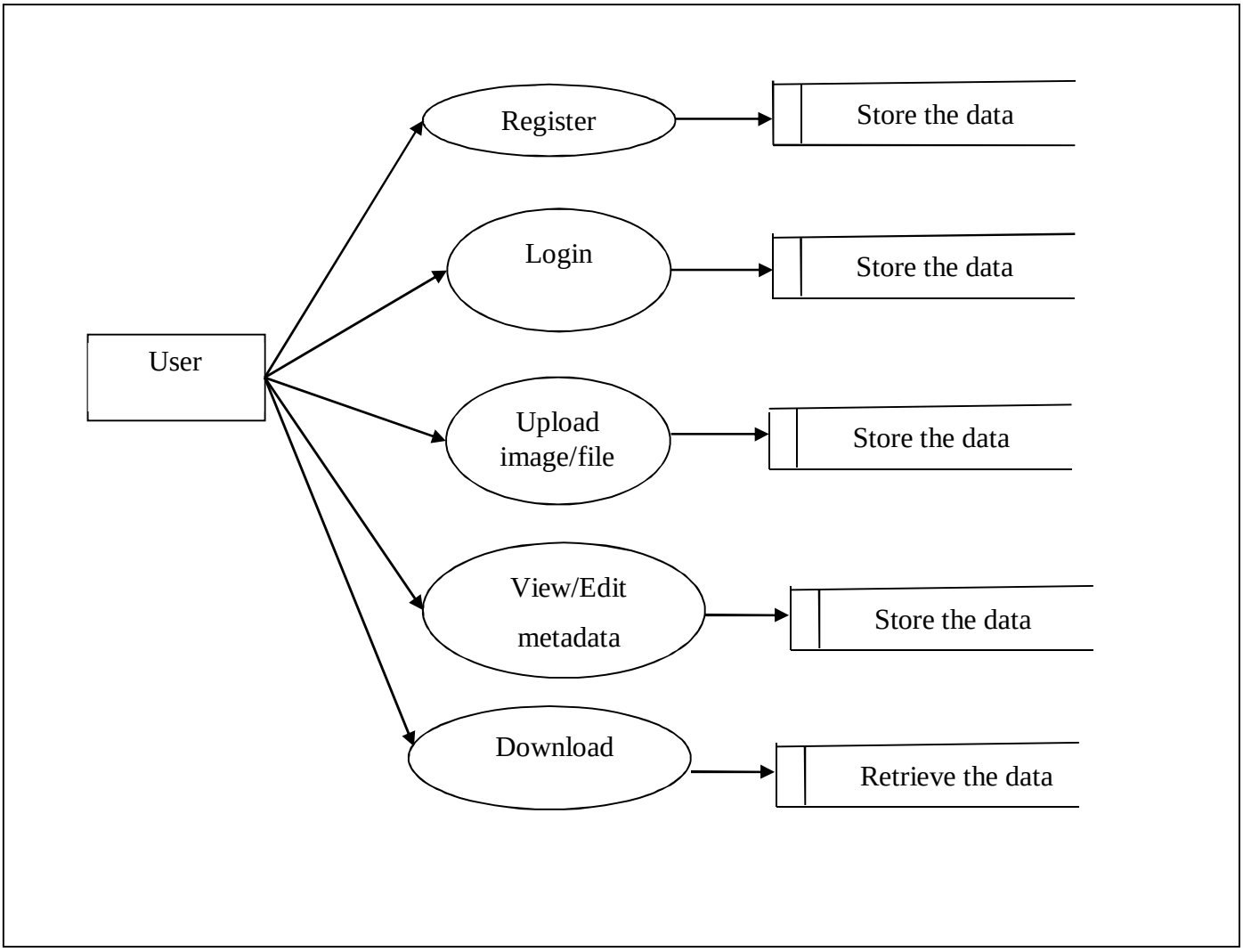


Fig 3. Storing of Data

2) UML Diagram

A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system. UML is a modern approach to modeling and documenting software. In fact, it's one of the most popular business process modeling techniques. It is the diagrammatic representation of software components. There are several types of UML diagram.

3) Usecase Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

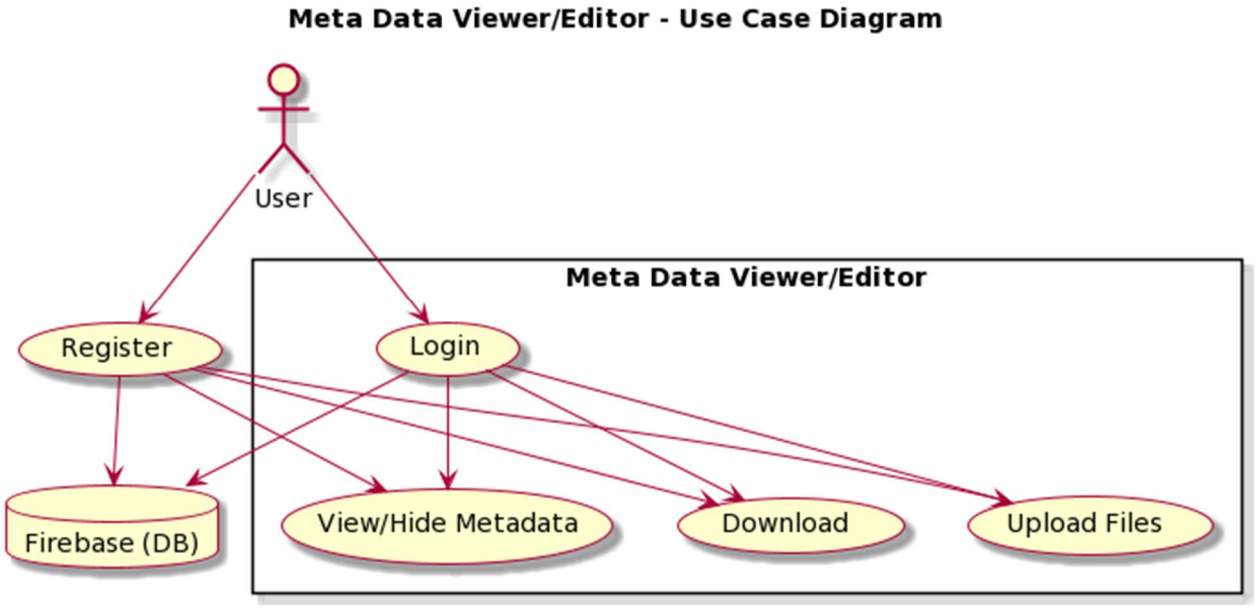


Fig 4. Usecase Diagram

4) Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

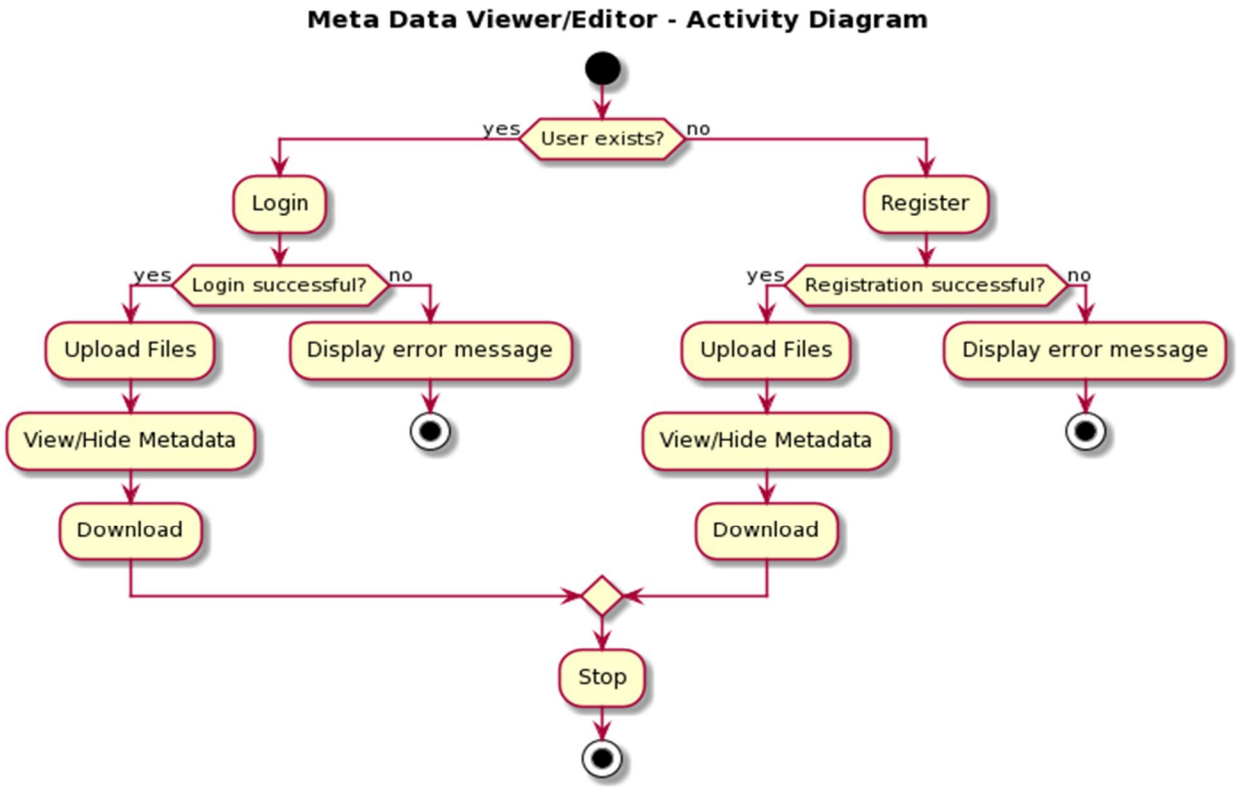
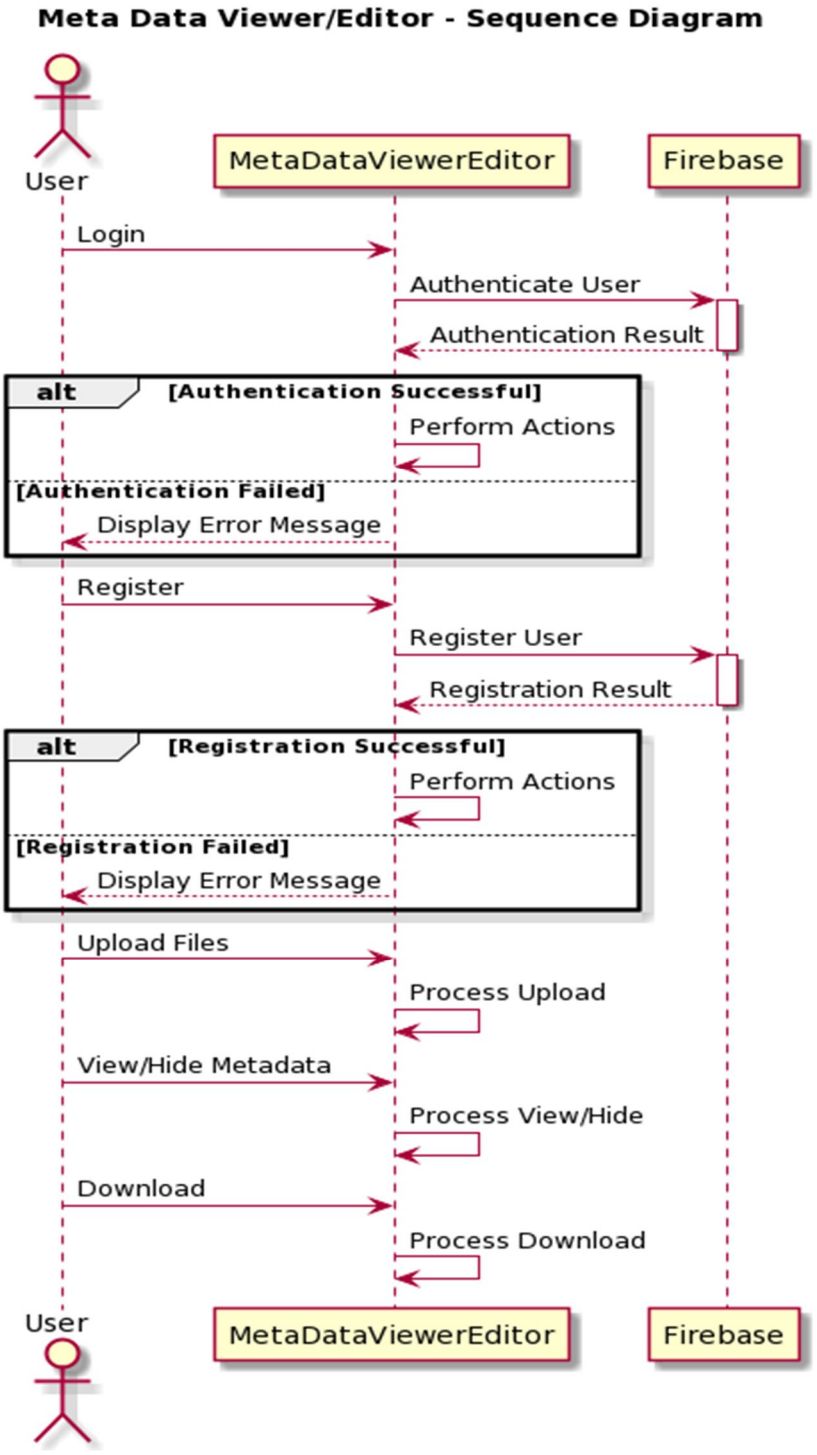


Fig 5. Activity Diagram

5) Sequence Diagram

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems. Sequence diagrams use notations like actor, lifeline and messages. Messages are of different types as follows create message, delete message, self-message and reply message. Sequence diagrams visualise how messages and tasks move between objects or components in a system.



ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

6) Class Diagram

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

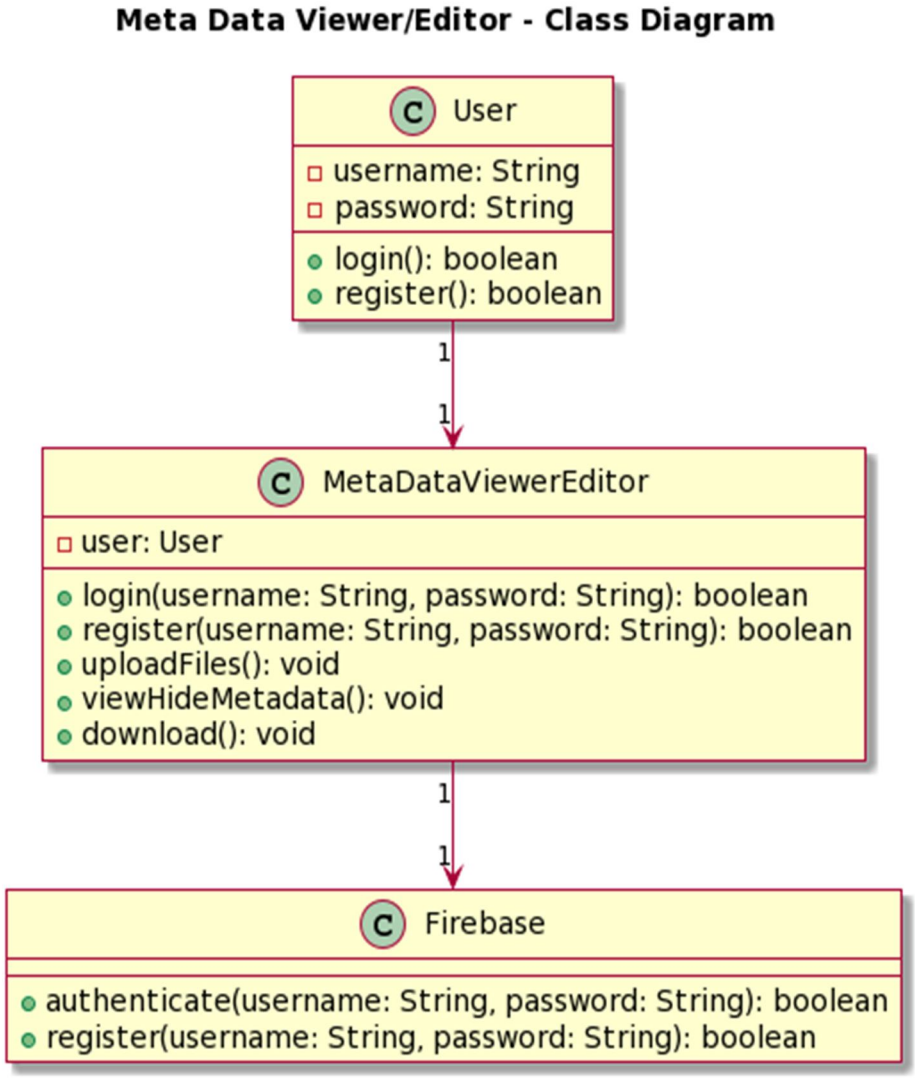


Fig 6. Class Diagram

IV. DESIGN & MODULE DESCRIPTION

Systems implementation is the process of: defining how the information system should be built (i.e., physical system design), ensuring that the information systemic operational and used, ensuring that the information system meets quality standard (i.e., quality assurance). Implementation is the process that actually yields the lowest-level system elements in the system hierarchy (system breakdown structure). System elements are made, bought, or reused. Production involves the hardware fabrication processes of forming, removing, joining, and finishing, the software realization processes of coding and testing, or the operational procedures development processes for operators' roles. If implementation involves a production process, a manufacturing system which uses the established technical and management processes may be required. The purpose of the implementation process is to design and create (or fabricate) a system element conforming to that element’s design properties and/or requirements.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

A. Modules

- Login
- Register
- Dashboard
- Edit Meta-data
- Result

B. Module Description

Login: - Users can login using the login page
Register: - Users can register using the register page
Dashboard: - They can upload their file in Dashboard
Edit Meta-data: - After uploading they can Edit the meta-data inside the uploaded file
Result: - Result page contains the final edited file, they can download the new file

V. IMPLEMENTATION

A. Front End

1) HTML

The HyperText Markup Language, or HTML(HyperText Markup Language) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

2) CSS

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs,variations in display for different devices and screen sizes as well as a variety of other effects.

B. BACK END

1) Flask

Flask is a lightweight, open-source web framework for building web applications in Python. It was developed to be easy to learn and use, and to provide developers with flexibility and control over their web applications. Flask provides a simple and easy-to-use interface for creating web applications, and it is ideal for building small to medium-sized web applications, APIs, and prototypes. Flask is a popular choice for building web applications in Python due to its simplicity, flexibility, and ease of use. It is well-documented and has a large and active community, which makes it easy for developers to find help and resources when building their web applications

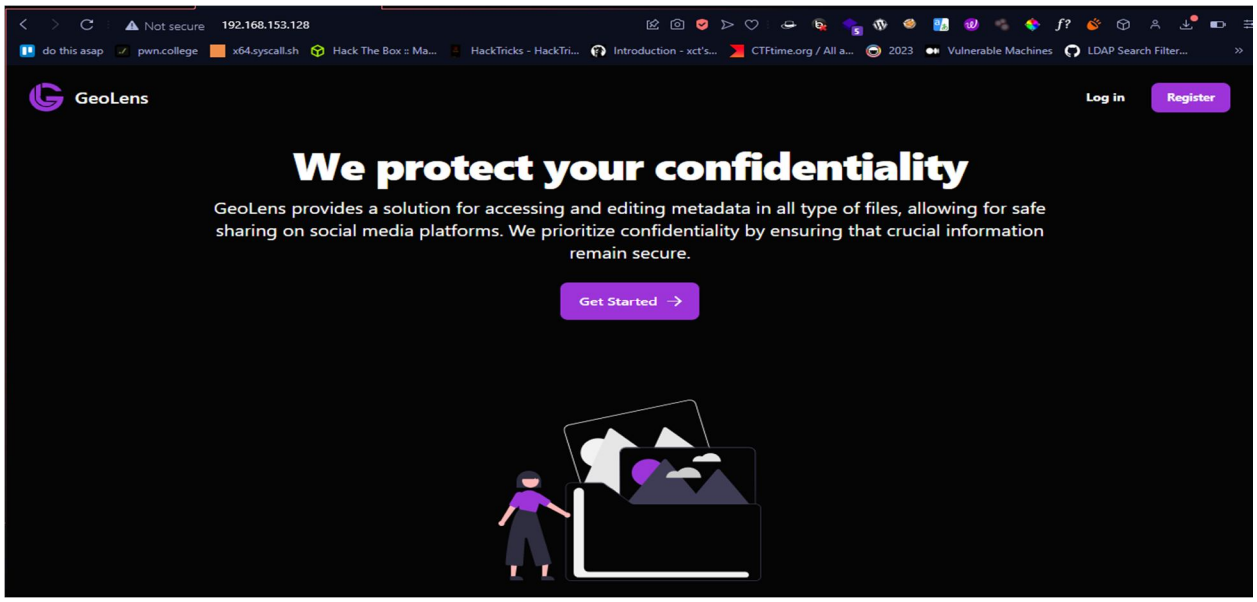
2) Firebase

Firebase is a mobile and web application development platform that provides a suite of backend services and tools for building and managing apps. It was originally founded in 2011 by Andrew Lee and James Tamplin and was later acquired by Google in 2014.Firebase offers a variety of services that developers can use to build their applications, including real-time database, authentication, hosting, cloud storage, and analytics. These services are designed to be easy to use and integrate seamlessly with other Google services, such as Google Cloud Platform and Google Analytics. One of the core features of Firebase is its real-time database, which allows developers to store and sync data in real-time across multiple clients.

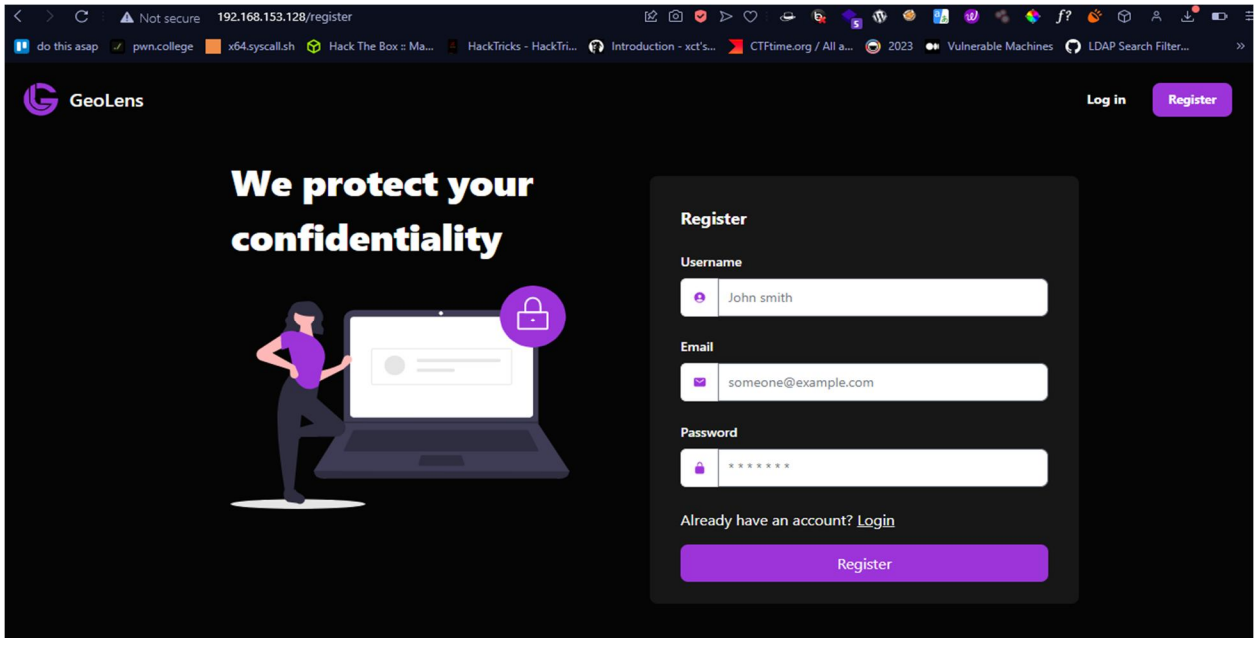
©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

This is particularly useful for applications that require real-time updates, such as chat apps or collaborative tools.Firebase also includes authentication features, allowing developers to add user authentication and authorization to their applications with just a few lines of code. Firebase supports several authentication methods, including email and password, social media logins, and phone number verification.Firebase hosting provides a simple way for developers to deploy their web apps and static content to the web. Firebase hosting is backed by a global CDN, which ensures fast and reliable delivery of content to users around the world.Finally, Firebase provides powerful analytics tools that allow developers to track user engagement, retention, and other important metrics. Firebase analytics integrates seamlessly with other Firebase services, making it easy to track how users are interacting with your app. Overall, Firebase is a powerful tool for developers looking to build scalable, reliable, and feature-rich web and mobile applications. With its easy-to-use services and powerful analytics tools, Firebase makes it easy for developers to focus on building great user experiences.

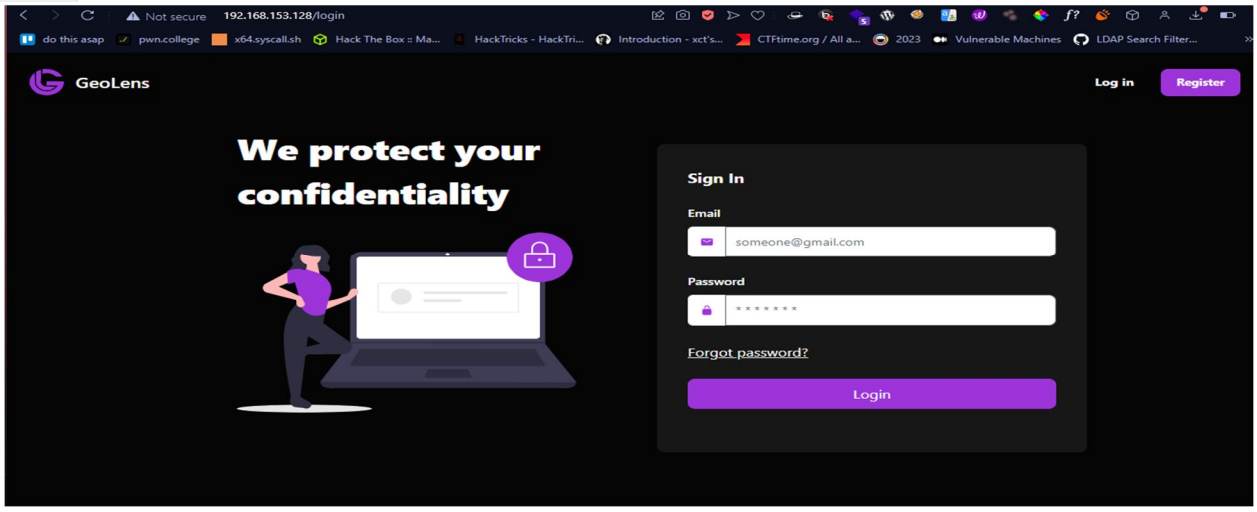
C. Screenshots



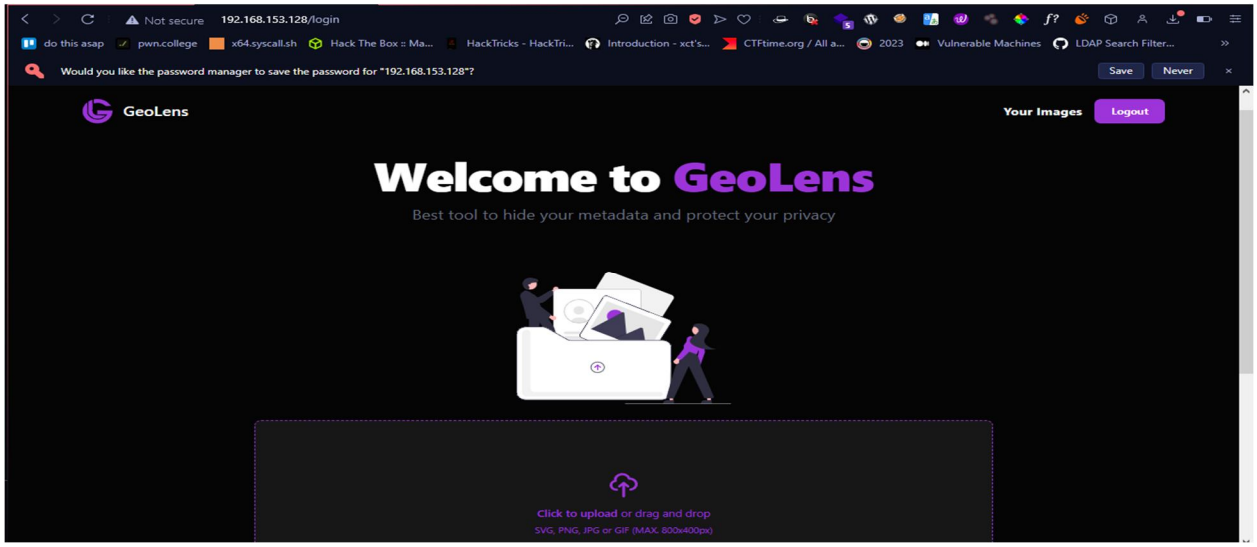
Home page



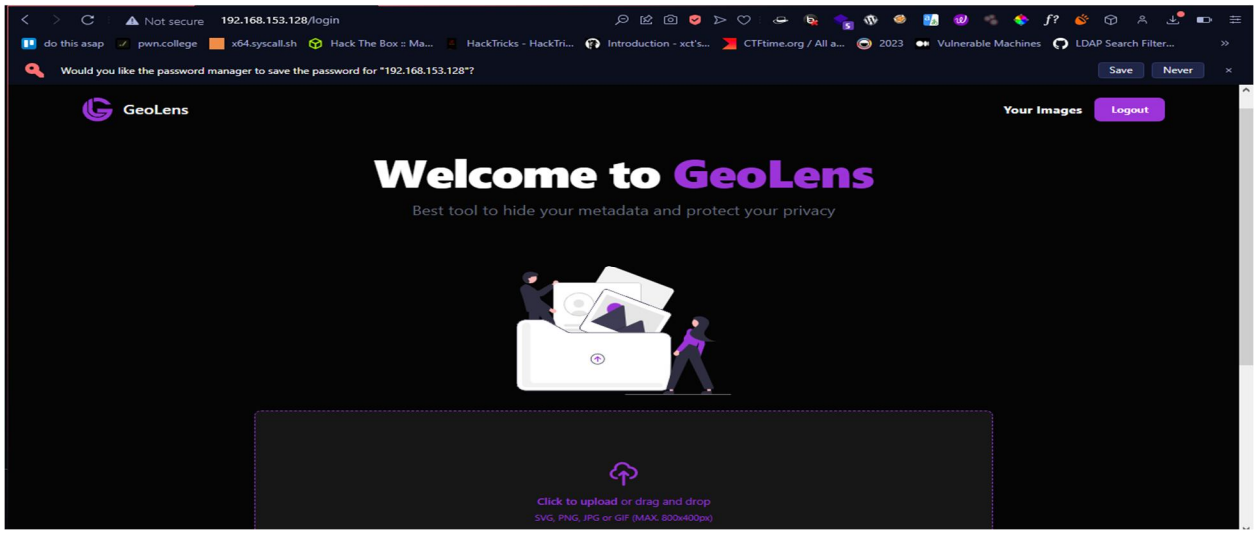
Register page



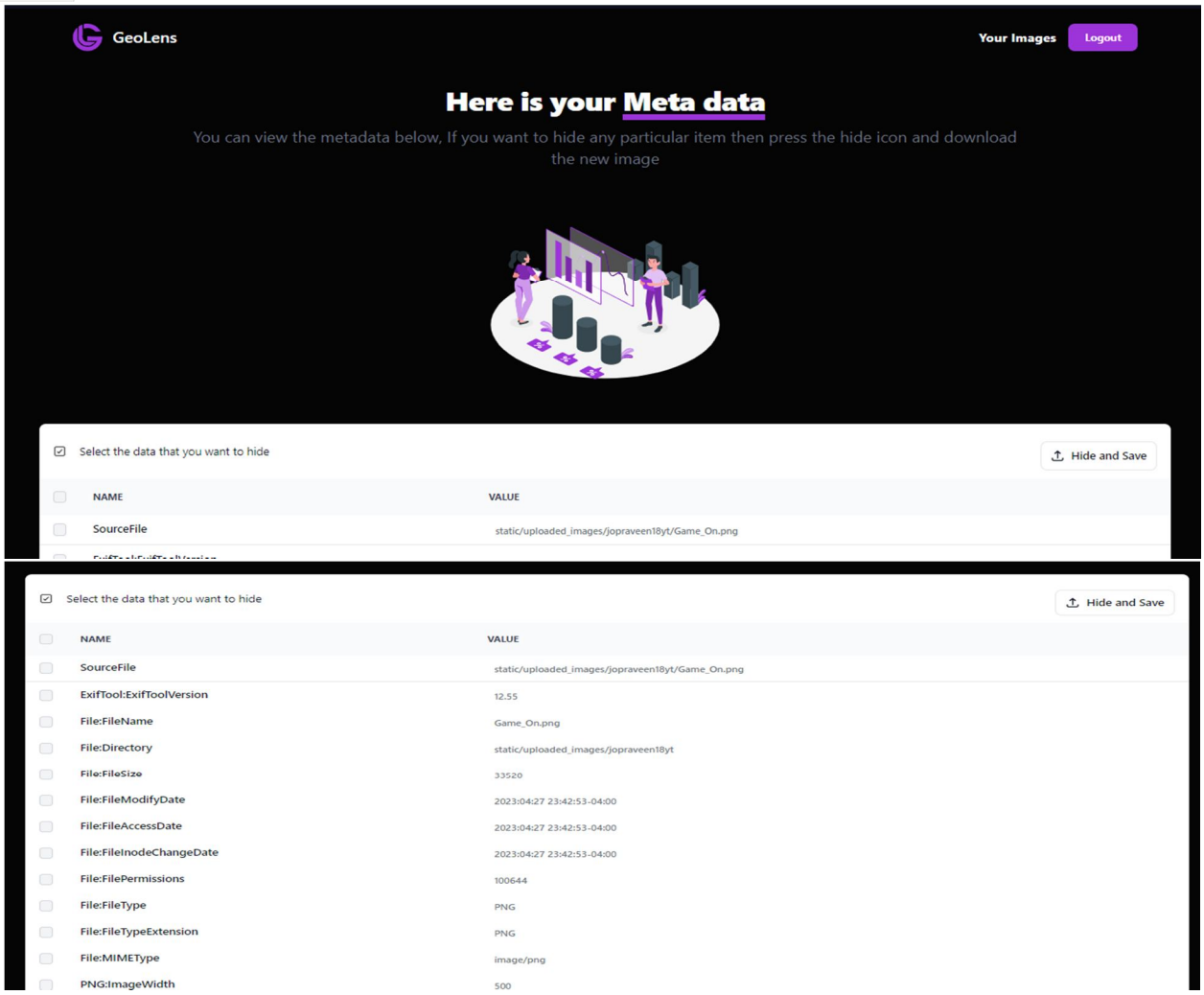
Login page



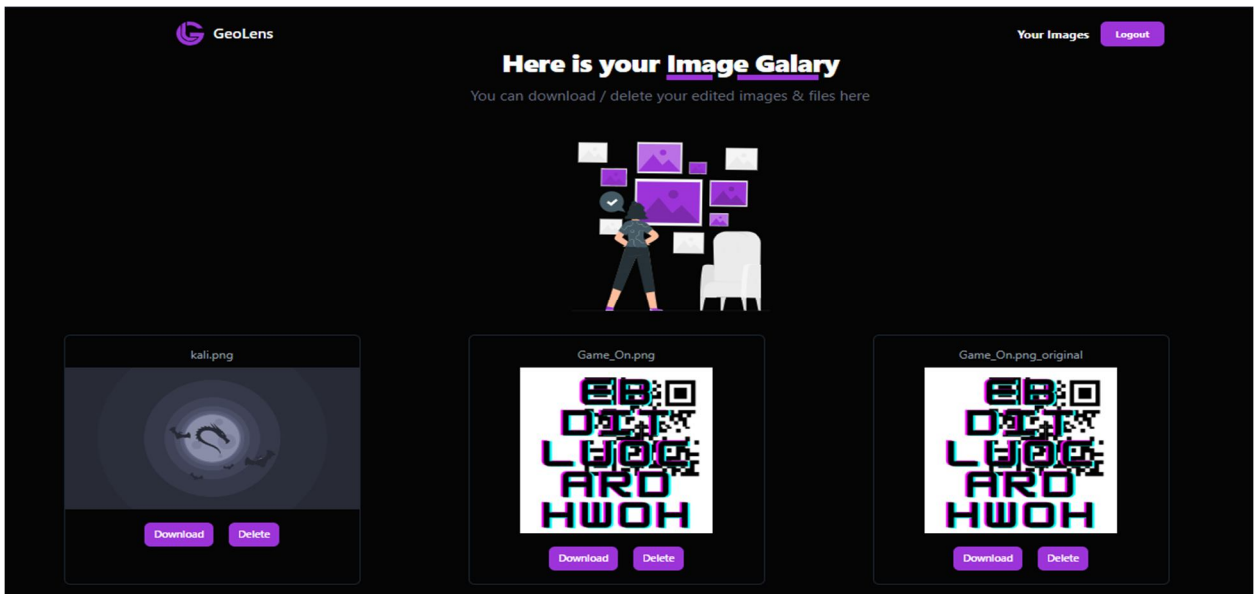
Dashboard



Upload page



Result page



Uploaded images

VI. TESTING

A. System Testing

System Testing is the testing of a complete and fully integrated software product. Usually software is the only one element of a larger computer based system. Ultimately, software is interfaced with other software/hardware systems. System Testing is actually a series of different tests whose sole purpose is to exercise the full computer based system. Software Testing is an important review of specification, design and coding. The increasing visibility of software as a system element and costs associated with the software failure are motivating forces for well-planned through testing.

Though the test phase is often thought of separate and distinct from the development effort first developers and then testing is a concurrent process that provides valuable information for the development team. There are at least three options for integrating project builder into the test phase.

- Testers do not install project builder, use project builder functionality to compile and source-control the modules to be tested and hand them off to the tester, whose process remains unchanged.
- The testers import the same project or the project that the developer uses.
- Create the project based on the development project but customized for the testers (For example- It does not include support documents, source) who imports it.

1) Testing Objectives

There are several rules that can serve as testing objectives. They are,

- Testing is executing a program with the intent of finding an error.
- A good test case is one that has a high probability of finding an undiscovered error.
- A successful test is one of that uncovers the undiscovered error.

If testing is conducted successfully according to the objective stated above, It will uncover the error in the software.

2) Types Of Testing

Testing is the process of executing the program with the intent of finding errors. Testing cannot show the absence of defects, It can only show that software errors are present. The testing principles used are

- Tests are traceable to customer requirements.
- 80% of errors will likely be traceable to 20% of program modules.
- Testing should begin ‘in-small’ and progress towards testing ‘in-large’.

The types of testing are

- Unit testing
- Integration testing
- White box testing
- Black box testing

In our project, we used unit testing and validation testing for checking the system.

B. Unit Testing

Unit testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

In our system, we tested each module as a separate unit. Each program from the modules are tested and corrected.

C. Validation Testing

Validation Testing ensures that the product actually meets the client's needs. It can also be defined as to demonstrate that the product fulfils its intended use when deployed on appropriate environment. Validation Testing is carried out as a whole system to ensure the system/product works well and that meets the requirements of the customer. We validated our project by doing this validation testing. It gives the desired result and satisfies the customer need.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

VII. CONCLUSION AND FUTURE WORK

Many users are unaware of the importance of metadata in image files, which can contain critical information such as the location, date, and time of the photo's capture, as well as camera settings and other relevant data. This information can be essential for organizing and categorizing images, as well as for legal and security purposes. However, most users lack the tools and knowledge required to view and edit this metadata, which can lead to privacy and security concerns when sharing images online. Hence This project will be very useful for maintaining confidentiality in social medias.

A. Future Enhancement

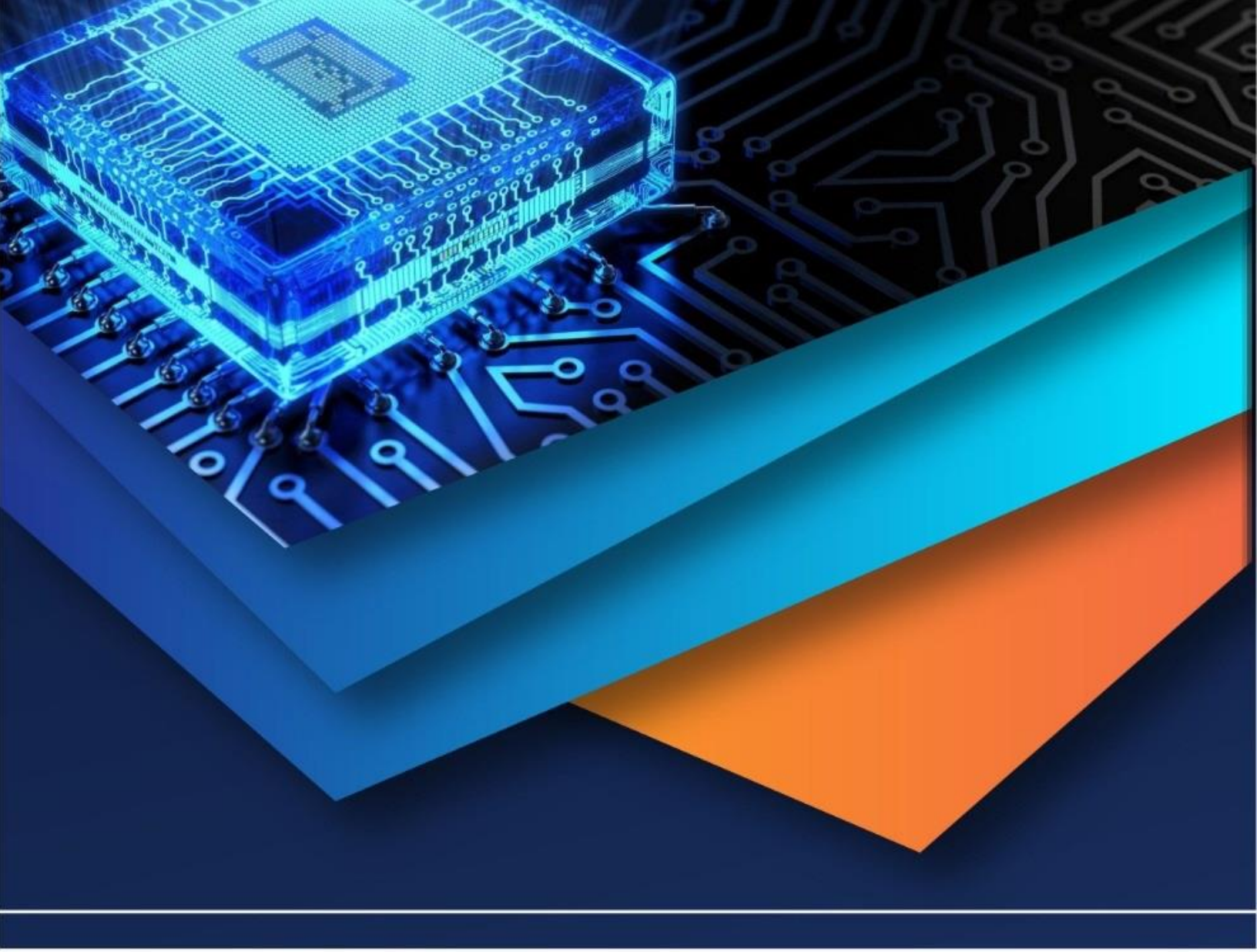
1) To implement Oauth2 to save files in the user's google drive

2) To create a mobile application for this web application, so users can easily upload/edit their files

REFERENCES

[1] Shu, X., Zhang, X., & Kankanhalli, M. S. (2017). A Survey on Metadata Extraction Techniques for Digital Images. ACM Computing Surveys (CSUR), 50(4), 58.
[2] Fu, H., Wang, M., Xie, X., & Zhang, H. (2018). Meta-Vision: Deep Visual Metadata Extraction and Mining. IEEE Transactions on Image Processing, 27(7), 3472-3486.
[3] Jiang, L., Lin, S., Cui, H., & Huang, Q. (2020). Metadata Extraction and Annotation for Images: A Survey. IEEE Transactions on Multimedia, 22(1), 83-98.
[4] Petrovic, M., & Radenkovic, M. (2016). A Comparative Analysis of Metadata Extraction Tools for Digital Image Collections. Information Systems, 61, 43-61.
[5] Sinha, S., & Aich, S. (2019). Efficient Image Metadata Extraction Using Deep Learning. Proceedings of the 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), 1-6.
[6] Zhang, Q., Wu, P., & Zhu, X. (2018). Metadata Extraction for Images Using Convolutional Neural Networks and Optical Character Recognition. In Proceedings of the 2018 3rd International Conference on Multimedia Systems and Signal Processing (ICMSSP), 1-6.
[7] He, R., Yin, X. C., Liu, W., & Wang, C. (2020). MetaExtractor: A Novel Deep Learning-Based Framework for Metadata Extraction from Images. IEEE Access, 8, 48006-48018.
[8] Chaves, R., Fan, J., & Wu, J. (2019). Deep Meta-Learning for Automatic Image Metadata Extraction. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), 3155-3164.
[9] Ding, J., Luo, W., & Carin, L. (2017). Deep Multi-Scale Representation Learning for Image Metadata Prediction. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS), 7230-7239.
[10] Ma, Y., Gao, Y., & Ji, X. (2019). Metadata Extraction for Images Based on Deep Learning and Transfer Learning. In Proceedings of the 2019 International Conference on Computer Information Science and Application Technology (CISAT), 179-183.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)