Advanced Python Programming -Lab Project

Team Members:

1) Aman Patel – 21BCP071
2) Parth Bhilwala – 21BCP073
3) Harsh Davra – 21BCP092
4) OM M Patel – 21BCP094
5) Madhuram Modi – 21BCP102

Project Title: Image Processing Website

Project Description:

The Image Processing Website is a feature-rich web application that allows users to upload, process, and analyse images.

Users can perform various image processing tasks, such as cropping, resizing, applying filters, background removal, and object recognition.

Additionally, the website provides a gallery where users can view all their uploaded images, edit images using different image processing functions, and analyse image metadata through data analytics and visualizations.

As an added functionality, the website supports web scraping to extract images from external websites, enhancing the user's image repository.

Project Features and Functions:

1. Image Upload:

Users can upload images to the website using drag-and-drop or file input. The uploaded images will be stored on the server.

2. Image Processing:

The website will automatically process uploaded images using Python's image processing scripts with OpenCV (cv2). Image processing tasks may include resizing, cropping, applying filters, or other transformations.

3. Background Removal:

The website will incorporate background removal functionality using machine learning models implemented with TensorFlow or PyTorch. This allows users to remove the background from their uploaded images.

4. Automated Image Processing:

The website will schedule and run Python scripts periodically to process images automatically. This ensures that all images in the gallery are regularly processed without manual intervention.

5. Image Gallery Display:

The website will display the processed images in an interactive image gallery using React components. Users can view the images and browse through them.

6. Image Metadata Storage:

Metadata for each image, such as the image name, upload date, and image location, will be stored in a MongoDB database. This allows for efficient image management and retrieval.

7. Data Analytics and Visualization:

The website will analyse image-related data, such as the number of uploaded images, image sizes, or processing times. Python's Pandas library will perform data analysis, and Matplotlib or Plotly will generate visualizations.

8. Automated Data Analytics:

Python scripts will be scheduled and run periodically to perform data analytics on the image metadata. This ensures that image-related insights and statistics are regularly updated.

9. User Interface:

The frontend, built using React, will provide a user-friendly interface for image upload, gallery display, and accessing data analytics insights.

10.Web Scraping :

If you choose to incorporate web scraping in the project, the website may fetch additional images or data from external sources for processing or analysis.

11. File Input/Output (I/O):

The website will utilize Python's file I/O to handle image processing tasks and store processed images in specific directories on the server.

12. User Authentication:

If user accounts and authentication are implemented, the website may allow registered users to manage their uploaded images and access personalized features.

13. Automated Image Retrieval:

The website may automatically retrieve and process images from external sources or social media platforms using web scraping.

14. Deep Learning Models:

The website will use TensorFlow or PyTorch to implement deep learning models for tasks such as image segmentation, image style transfer, or object recognition.

15. Image Style Transfer:

Users may have the option to apply style transfer to their uploaded images, using pre-trained models to adopt the style of famous artworks.

16. Object Recognition:

The website can employ object recognition models to identify and label objects within the uploaded images.

By offering these capabilities, the Automated Image Processing and Gallery website becomes a comprehensive platform that allows users to interact with their images in unique and exciting ways.

It enables them to not only manage and process their images but also perform advanced tasks such as background removal and style transfer through deep learning models.

Roadmap to the Project:

1.Project Setup:

- Set up a new project folder with separate frontend and backend directories.
- Initialize a new React app in the frontend directory using Create React App.
- Create a new Node.js project in the backend directory using npm or yarn.

2.Frontend Development:

- Create the Home page component with a welcoming message and a "Get Started" button.
- Build the Upload page component with an area for drag and drop, an "Upload Image" button, and an "Edit" button to navigate to the Edit Image page for the selected image.
- Develop the Edit Image page component with various function buttons on the left side and feature areas on the right side based on the selected function. Display the uploaded image in the remaining area on the right side.

3. Backend Development:

- Set up a MongoDB database to store image metadata, including file names and timestamps.
- Implement backend API endpoints to handle image upload, image retrieval, and image metadata storage.
- Integrate Python scripts for background removal and object recognition. Set up API endpoints to trigger these tasks for specific images.

4. Gallery Page:

- Create the Gallery page component to display a grid of all uploaded images.
- Implement functionalities to view individual images and navigate to the Edit Image page for each image.

5. Image Processing Functionality:

- Implement image processing tasks, such as cropping, resizing, and applying filters, using JavaScript libraries like Canvas API or third-party image processing libraries.

6. Background Removal and Object Recognition:

- Integrate the Python scripts for background removal and object recognition into the backend.

- Set up API endpoints to call the Python scripts for these tasks on uploaded images.

7. Data Analytics Page:

- Develop the Data Analytics page to analyse and visualize image metadata using Python's Pandas and Matplotlib libraries.

8. User Authentication and Authorization:

- Implement user authentication and authorization using libraries like Passport.js for Node.js and JWT (JSON Web Tokens).

- Set up secure API endpoints for user-specific functionalities.

9. User Management:

- Add functionalities for users to manage their uploaded images, edit image information, and delete images if necessary.

10. User Interface Enhancements:

- Improve the user interface and user experience by adding animations, transitions, and responsive design to the website.

11. Testing and Bug Fixing:

- Thoroughly test the website on different devices and browsers.

- Identify and fix any bugs or issues that may arise during testing.

12. Deployment:

- Deploy the fully developed and tested website to a web hosting service or cloud platform of choice to make it accessible to users online.

13. Documentation:

- Create comprehensive documentation for the project, including setup instructions, API documentation, and explanations of how various functionalities work.

Conclusion:

The Image Processing Website is a comprehensive web application that allows users to upload, process, and analyse images. It incorporates both frontend and backend components, image processing with Python, user authentication, and data analytics functionalities. By following the above roadmap and continuously improving the project, the Image Processing Website will be a powerful tool for users to manipulate and analyse their images online.