International Journal on Science and Technology (IJSAT)
E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

Image Resize with Live Capture

Poovarasan A

Postgraduate Student(MCA)
DepartmentofComputerApplications
Dr.M.G.R.Educational and Research Institute ,Chennai,India san9629565074@gmail.com

Abstract:
This project focuses on the development of a system that captures live images through a webcam or camera device and resizes them dynamically based on user-defined parameters or predefined dimensions. The core objective is to automate the image acquisition and resizing process, making it suitable for applications such as identity verification, online forms, image preprocessing in machine learning, and real-time visual content editing.

The system utilizes a live feed from a camera to capture frames in real time. Upon capturing an image, it processes the frame using image processing libraries (such as OpenCV or PIL) and resizes it while maintaining the aspect ratio or adjusting to custom dimensions as specified. The interface allows users to preview the resized output instantly and optionally save the processed images in various formats.

Keywords: An Realtime Image Resize to Between In The Live Camera Capture.

1. INTRODUCTION:

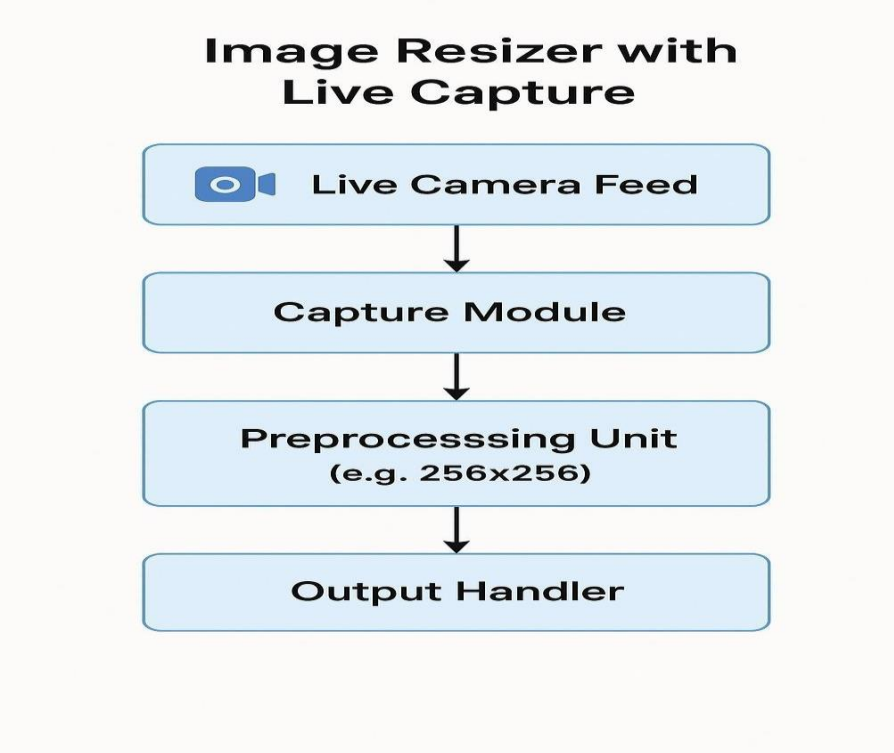
In the digital age, images play a crucial role in communication, documentation, and data processing. Whether used for social media, official documentation, machine learning datasets, or online forms, images often need to be resized to meet specific requirements related to dimensions, resolution, or file size. Manual resizing of images is time-consuming and inefficient, especially when working with large volumes or needing immediate feedback.To address these challenges, this project proposes a system that integrates live image capture with dynamic resizing capabilities. The system leverages a real-time camera feed to capture images instantly and allows users to resize them according to predefined or custom parameters. This eliminates the need for separate steps of capturing, uploading, and editing images, thereby streamlining the workflow.

The integration of image processing tools such as OpenCV or PIL (Python Imaging Library) enables real-time resizing while preserving the quality and aspect ratio of the image. Furthermore, the system is designed to be user-friendly, offering an intuitive interface where users can capture, preview, resize, and save images with minimal effort.This project has practical applications in various domains such as online registration portals, automated ID photo generation, educational institutions, passport or license processing systems, and anywhere real-time image handling is required.

2. LITERATURE REVIEW/RELATED WORK

Image capture and resizing are fundamental tasks in image processing, widely used in various applications ranging from photography to biometric systems and web development. Over the years, numerous techniques and tools have been developed to automate and enhance these processes. Live image capture has been extensively implemented using digital cameras and webcams integrated into software applications. Open-source libraries like OpenCV have become standard tools in the computer vision community for real-time image capture and processing. Research has shown that OpenCV's integration with programming languages like Python and C++ provides a robust and flexible platform for implementing live feed functionalities.

3. SYSTEMDESIGN



3.1. Architecture Diagram

Figure3.1: ArchitectureDiagram

3.2. DataFlowDiagram(DFD)

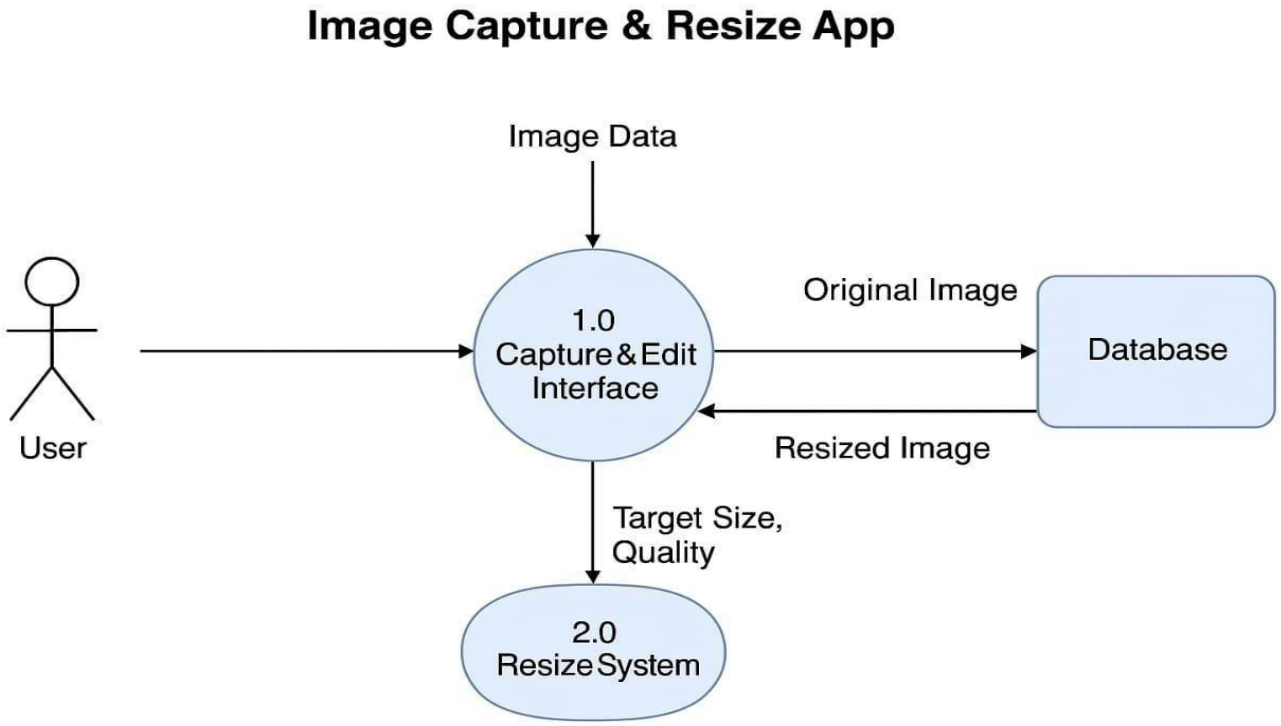


Figure3.2: DataFlowDiagram

3.3. UseCase Diagrams

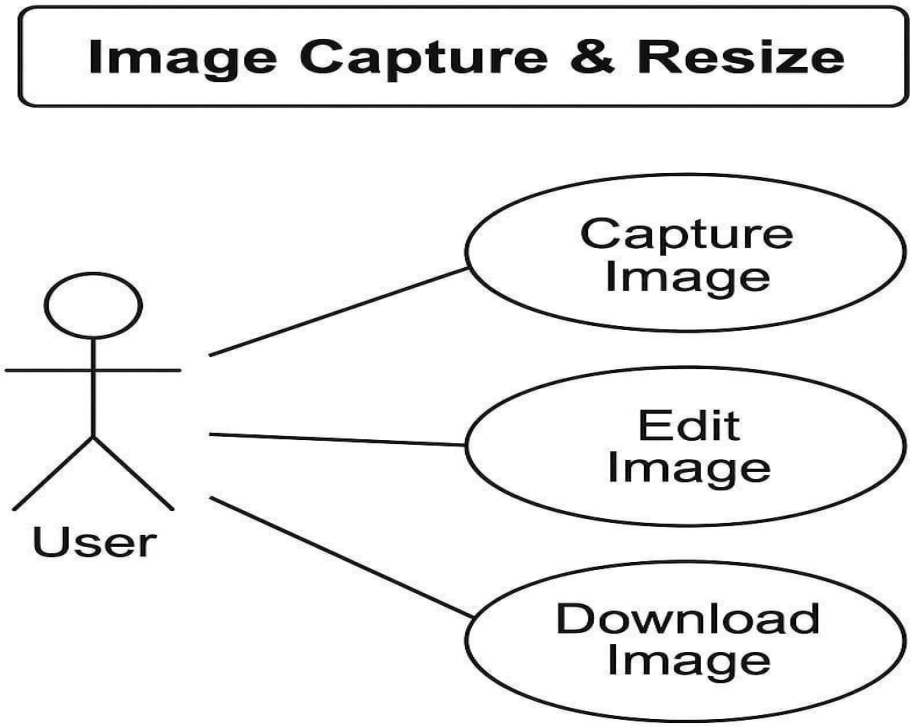


Figure3.3:UseCase Diagram

3.4. SequenceDiagram

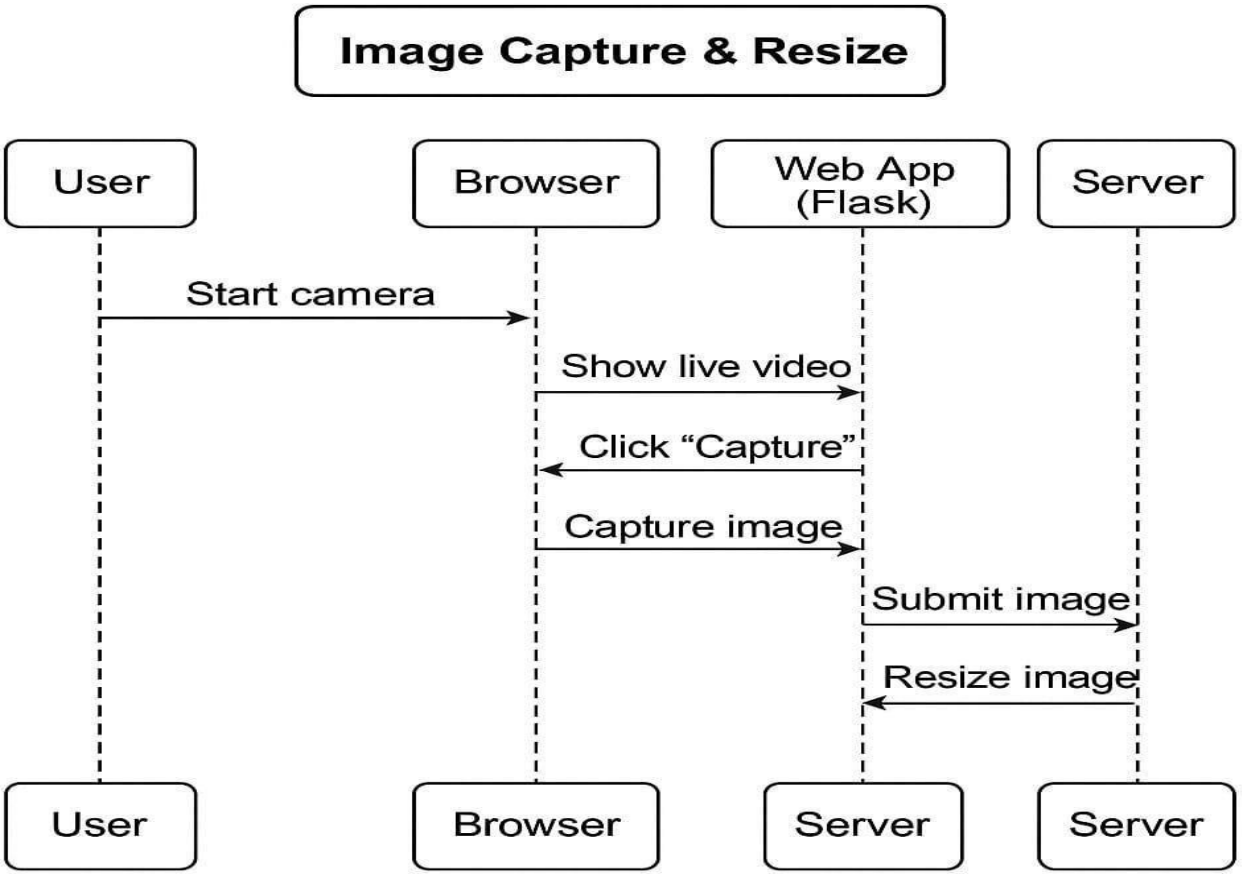
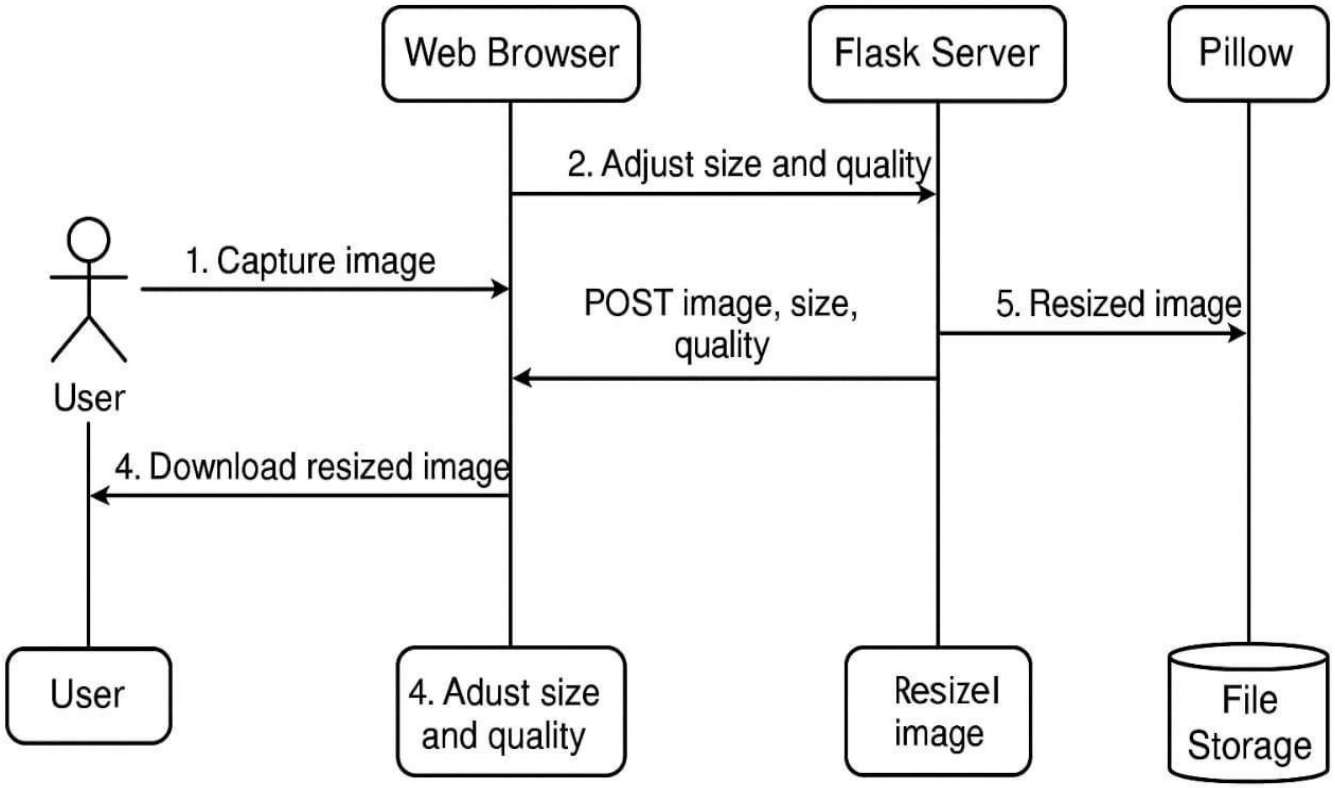


Figure3.4: SequenceDiagram

3.5. CollaborationDiagram

Figure3.5:CollaborativeDiagram



3.6. DatabaseDesign

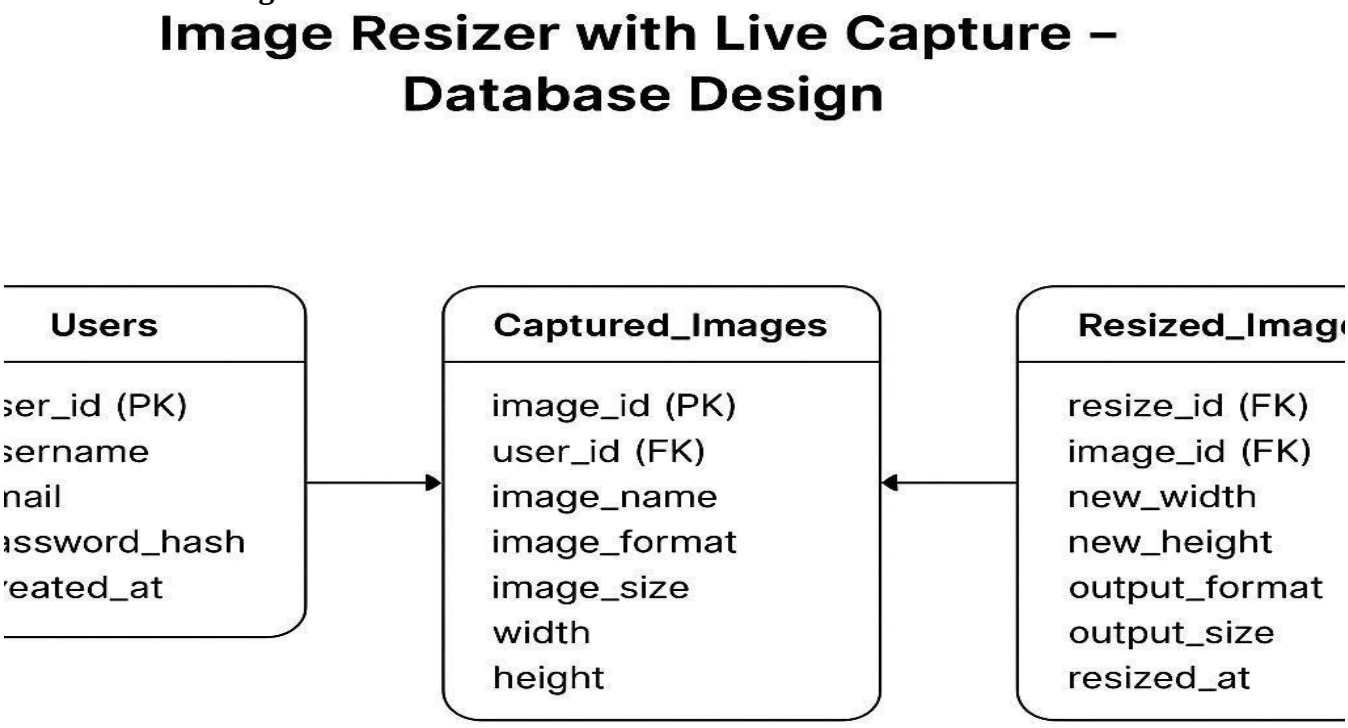


Figure3.6:DatabaseDiagram

4. RESEARCH METHODOLOGY

Test Case ID	Test Case Description	Input	ExpectedResult	Pass/Fail Criteria
TC001	Launchcameraand capture image	Click"Capture"button	Livecameraopens,and image is captured	Imagepreview displayedafter capture
TC002	Resize image to specific dimensions(e.g., 200x200)	Capturedimage, Resize:200x200	Imageisresizedcorrectly	Newimagematches specified size
TC003	Resizewithaspect ratio lock enabled	Captured image, Aspect Ratio: Locked	Image resizes proportionally	Nodistortion;aspect ratio preserved
TC004	Resizewithaspect ratiolockdisabled	Capturedimage, Resize:300x100	Imageresizestoexact size	Imageappears resizedevenif stretched
TC005	Saveresizedimage	Click"Save"after resize	Imagesavedtodeviceor path	Fileexistsatsaved location
TC006	Resizeimageusing percentage (e.g.,50%)	Capturedimage, Resize: 50%	Imageishalforiginalsize	Dimensionsreduced correctly
TC007	Capturefailsdueto camerapermissions	Blockcamera access	Showerrormessage	Appropriatealertor message shown
TC008	Resize with invalid input(e.g.,negative size)	Resize:- 100x200	Error/Validationshown	Invalidsizenot allowed
TC009	Resizehighresolution image (e.g.,4K)	Capturehighres image	Resizingworks correctly	Systemdoesn'tcrash or lag

TC010	UIelementsrespond during live preview	Startlive preview	Allbuttons functional	NofreezingorUI lockup

Table5.1: SampleTestCase

This methodology ensures a practical, user-centered, and technically sound development process, resulting in a reliable and efficient system for live image capture and resizing.

In this initial phase, the specific needs and objectives of the system were identified. These included:

- Capturing images using a live camera feed.
- Resizing captured images based on user-defined or predefined dimensions.
- Maintaining image quality and aspect ratio.
- Providing an interactive graphical user interface (GUI) for usability.
- Supporting image saving in various formats (e.g., JPEG, PNG).

This phase involved reviewing similar applications, gathering user expectations, and defining functional and non-functional requirements.

5. TESTING AND RESULTS

5.1 Testing

Thepurposeoftestingistodiscovererrors.Testingistheprocessoftrying todiscovereveryconceivablefaultor weaknessinaworkproduct.Itprovidesawaytocheckthefunctionalityofcomponents,sub – assemblies,assemblies and/or a finished product It is the process of exercising software with the intent of ensuring**.**

Software system meets its requirements and user expectations and does not fail in an unacceptable manner.There are various types of test. Each test type addresses a specific testing requirement.

5.2 TypesofTestingPerformed

•**UnitTesting:**Testedindividualcomponentsinisolation.

•**IntegrationTesting:**Verifiedinteractionsbetweendifferentmodules.

•**FunctionalTesting:**Validatedthecompleteapplicationagainstspecifiedrequirements.

6. SCREENSHOTS:

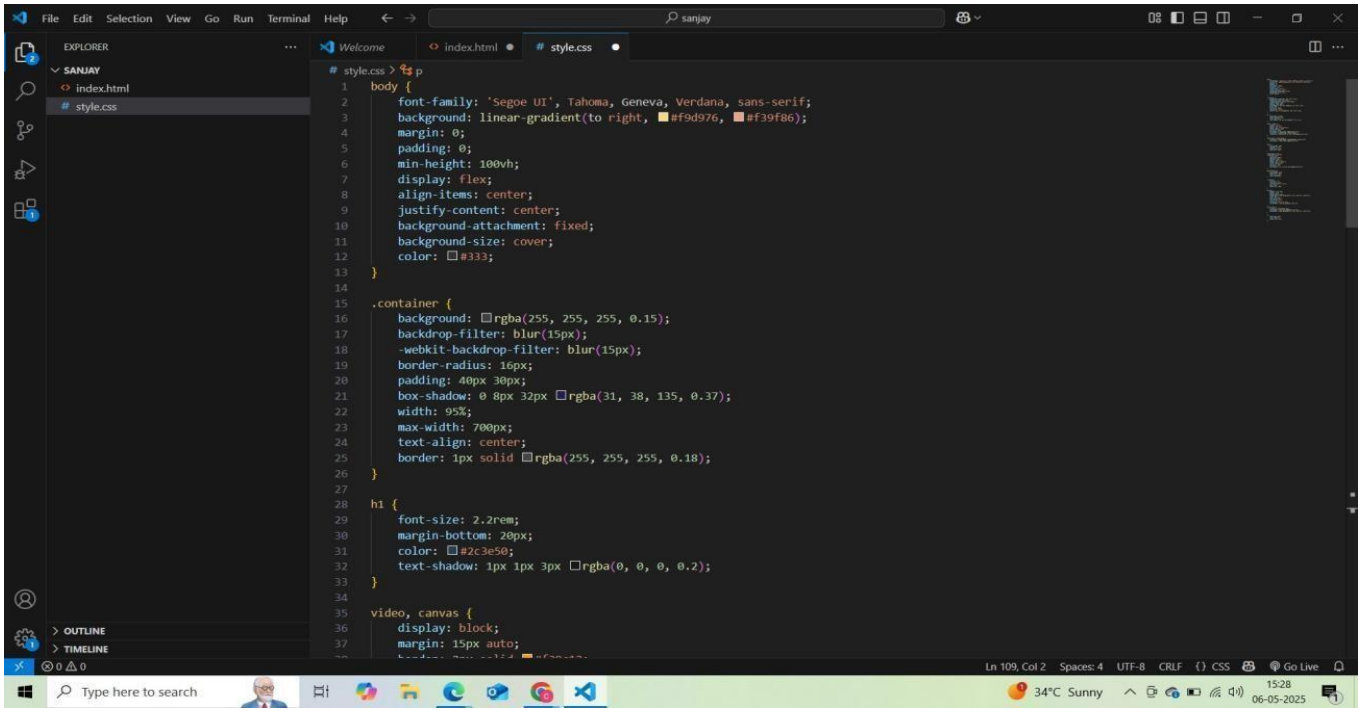


Figure 6.1

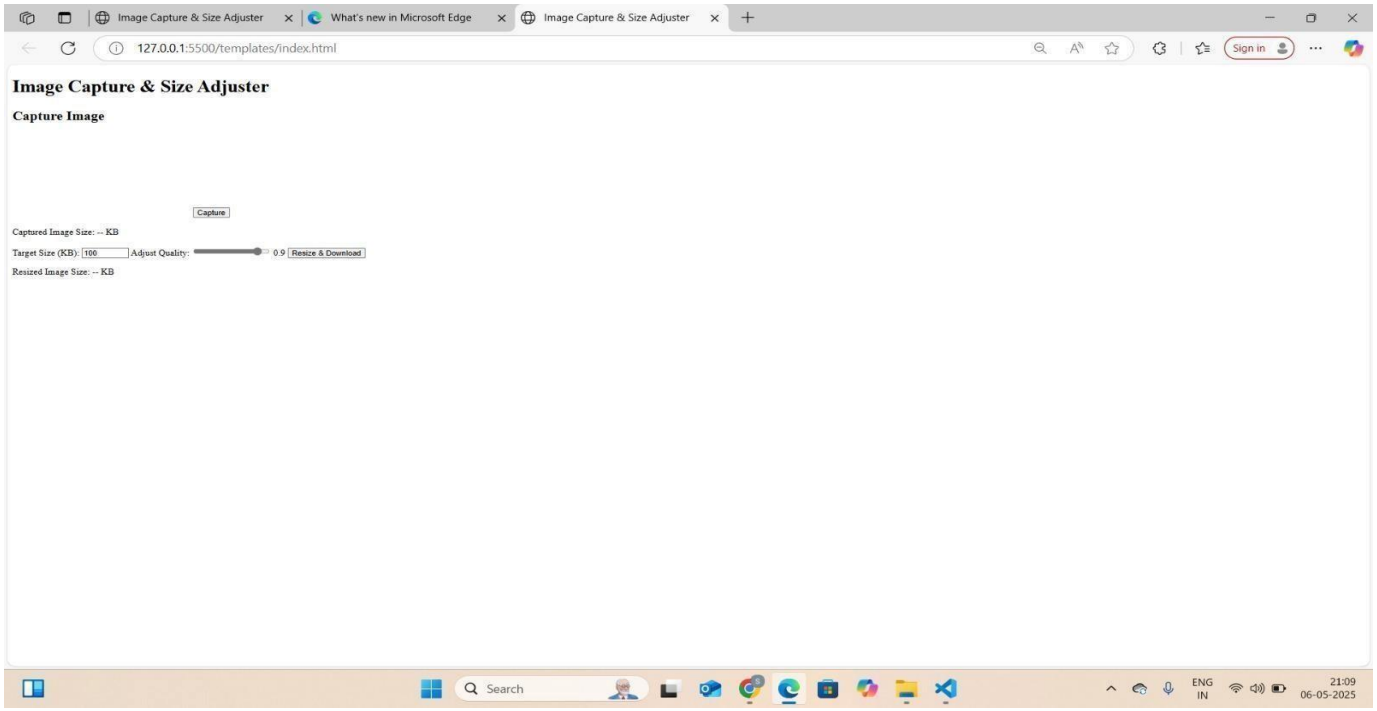


Figure 6.2

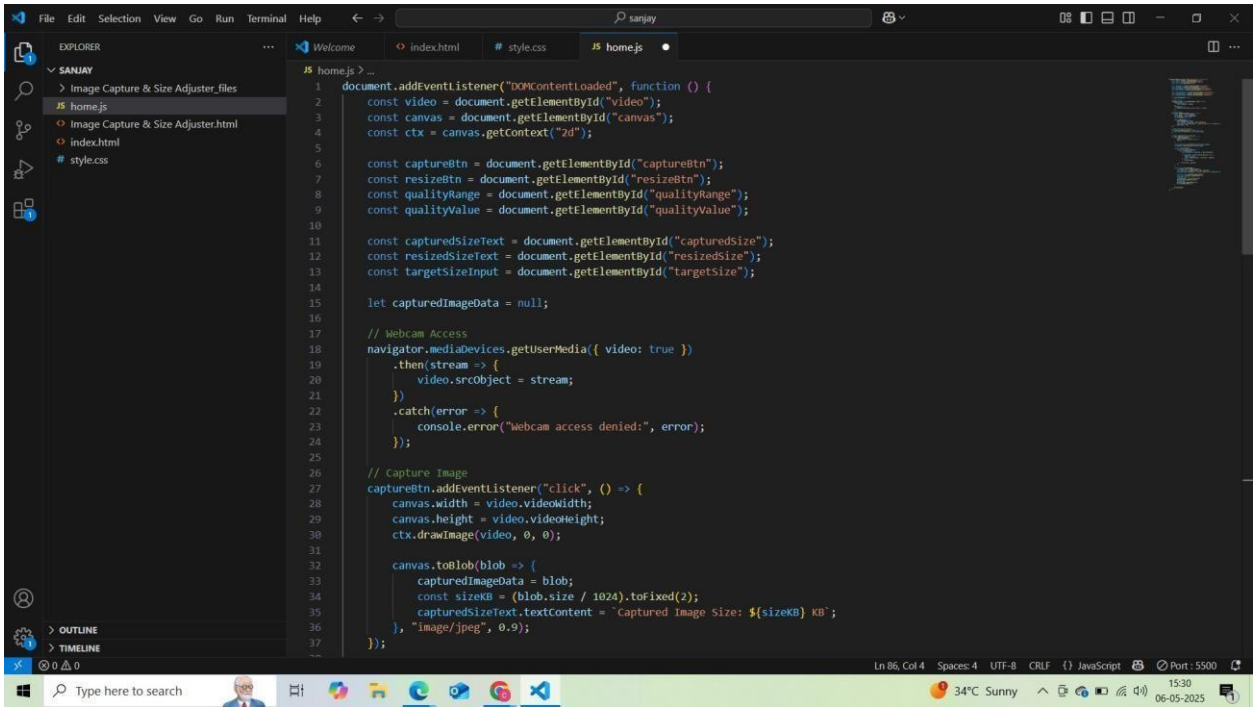


Figure 6.3

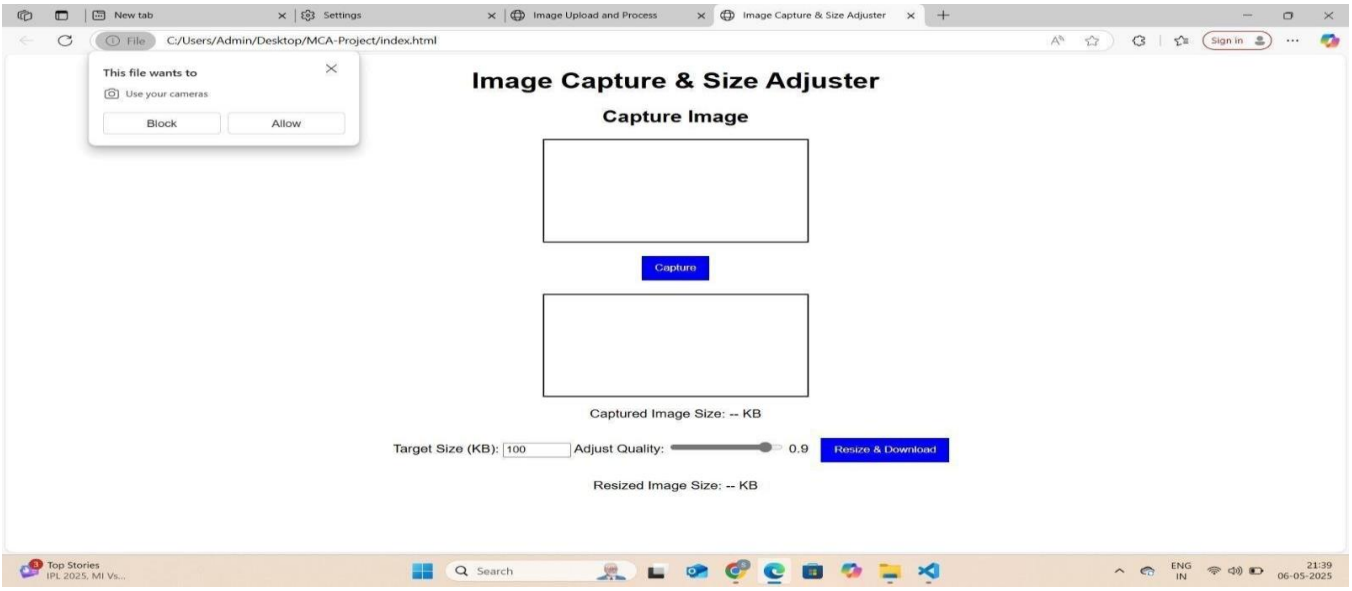


Figure 6.4

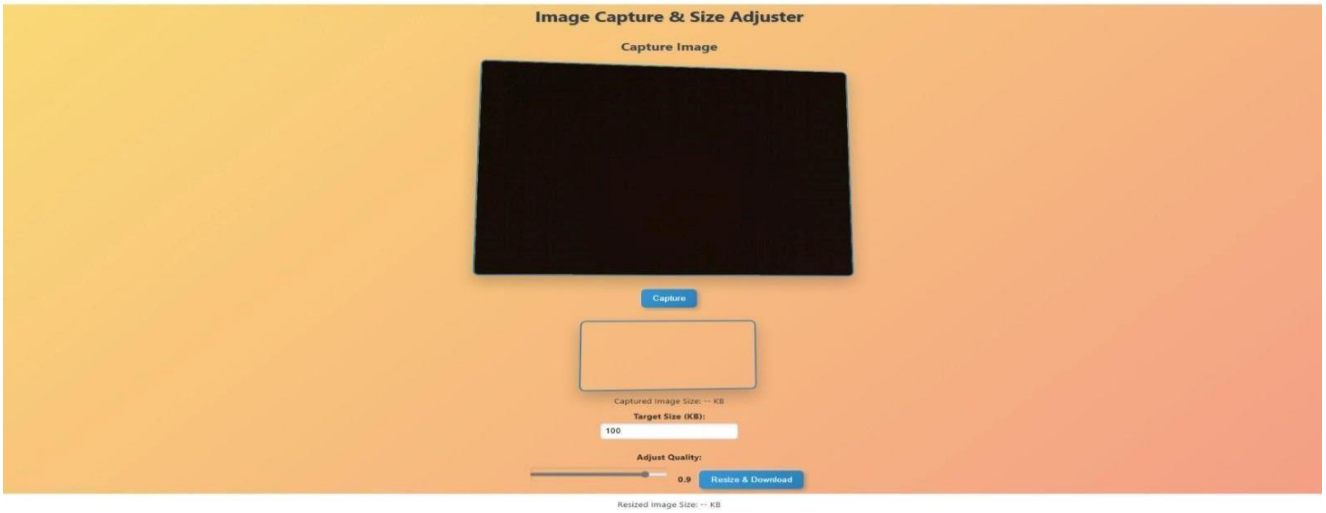


Figure 6.5

7. CONCLUSION

The Image Resizer with Live Capture system offers a comprehensive, user-centric solution for real-time image processing within the web environment. Designed with accessibility and ease of use in mind, the applicationallowsuserstocaptureimagesinstantlythroughtheirdevice'swebcam,resizethemtospecific dimensions—either custom or predefined—and download them in their desired format, all within a few simple clicks.This eliminates the need for external editing softwareor complex tools, makingthe process of image manipulation fast, straightforward, and accessible to users of all skill levels.

Thesystemisbuiltuponamoderntechnologystackthatensuresperformance,compatibility,andscalability. Thefront-end,developed usingHTML,CSS,JavaScript,andAngular, providesaresponsiveandinteractive user interface that performs smoothly across various devices and browsers. Angular's component-based architecture allows for modular design, which simplifies both development and maintenance. On the backend, Django serves as a robust and scalable framework that handles image processing logic, file management,andsecurecommunicationbetweentheclientandserver.Together,thesetechnologiescreate a cohesive system that delivers efficient processing and real-time feedback.

8. FUTUREENHANCEMENTS

ThecurrentimplementationoftheImageResizerwithLiveCapturesystemprovidesafunctionalinterface for capturing images via a camera module and resizing them to preset or user-defined dimensions.

However,thereareseveralpotentialimprovementsandadvancedfeaturesthatcanbeincorporatedin future versions to enhance usability, performance, and versatility. These include:

1. AI-PoweredImageOptimization
   Integrate machine learning models to automatically detect and preserve important content (e.g., faces,objects)whileresizing,ensuringminimallossofinformationandbetterimagecomposition.
2. BatchProcessingSupport
   Extend the system to allow users to capture and resize multiple images simultaneously, significantlyimprovingproductivityforusecasessuchasphotodocumentationorbulkmedia uploads.

3. Real-TimeAspectRatio Suggestions
   Implementanintelligentsuggestionenginethatrecommendsthebestaspectratiosbasedonthe captured content and target application (e.g., web, print, social media).

REFERENCES

1. Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools. https://opencv.org
2. Gonzalez, R. C., & Woods, R. E. (2018). Digital Image Processing (4th ed.). Pearson.
3. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems, 25, 1097–1105.
4. Zhang, K., Zuo, W., Chen, Y., Meng, D., & Zhang, L. (2017). Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. IEEE Transactions on Image Processing, 26(7), 3142–3155. https://doi.org/10.1109/TIP.2017.2662206
5. Lundh, F. (2001). Python Imaging Library (PIL) Handbook. PythonWare. https://pillow.readthedocs.io