PDP 2

IBM-CAD

TEAM LEAD : DHANA LAKSHMI S

TEAM M1 : KALPANA SRI R

TEAM M2 : JEEVITHA V

DEPARTMENT OF CSE

KARPAGAM INSTITUTE OF TECHNOLOGY,COIMBATORE

DETERMINE THE REQUIREMENTS

Customer Journey Maps

**Requirements for Image Resize Website:**

1. Image Upload:

- Users should be able to easily upload images from their devices or through integration with cloud storage services.

- Support for various image formats (e.g., JPEG, PNG, GIF) should be provided.

2. Image Resizing Options:

- Users should have the ability to specify desired dimensions or aspect ratios for the resized images.

- Options for percentage-based resizing, fixed width/height, and custom dimensions should be available.

- Preset resizing options for common use cases (e.g., social media profile pictures, website banners) can be included.

3. Image Quality and Compression:

- The system should optimize the resized images while maintaining acceptable quality.

- Compression algorithms should be implemented to reduce the file size without significant loss in image quality.

4. Batch Resize:

- Users should be able to select multiple images and resize them in a single batch operation.

- The system should handle the resizing process efficiently to avoid performance issues.

5. Preview and Comparison:

- Users should have the ability to preview the original and resized images side by side.

- A slider or interactive tool can be provided to compare different sizes and variations of the images.

6. Download Options:

- Users should be able to download the resized images in the desired format (e.g., JPEG, PNG) and quality.

- Provide options for individual or bulk download of the resized images.

7. User Authentication and Account Management:

- Implement user registration and login functionality to manage user accounts.

- Allow users to save their preferences, resize history, and manage their uploaded images.

8. User Interface:

- Design a user-friendly and intuitive interface with clear instructions and visual cues.

- Ensure responsive design to support various screen sizes and devices.

- Consider accessibility guidelines for users with disabilities.

9. Performance and Scalability:

- Optimize image processing and resizing operations for fast and responsive performance.

- Implement caching mechanisms to improve the speed of subsequent image resize requests.

- Design the system to handle a large number of concurrent users and resize requests.

Customer Journey Maps:

1. User Uploads and Resizes a Single Image:

- User lands on the website and is prompted to upload an image.

- User selects an image from their device or integrates with a cloud storage service.

- The system validates the image format and size.

- User specifies the desired dimensions or chooses a preset resizing option.

- The system processes the image, resizes it, and presents a preview of the resized image.

- User can make further adjustments if needed.

- User clicks on the download button to save the resized image.

2. User Batch Resizes Multiple Images:

- User lands on the website and is prompted to upload multiple images.

- User selects multiple images from their device or integrates with a cloud storage service.

- The system validates the images' formats and sizes.

- User specifies the desired dimensions or chooses a preset resizing option.

- The system processes each image, resizes them, and presents a preview of the resized images.

- User can make further adjustments if needed.

- User clicks on the download button to save the resized images as a zip file.

3. Returning User with Account Management:

- User logs in to their account on the website.

- User can access their previously uploaded images and resize history.

- User can upload new images for resizing.

- User can modify their account settings and preferences.

- The system provides personalized recommendations based on the user's past resizing patterns.

**REQUIREMENT ANALYSIS**

1. Functional Requirements:

a. Image Upload:

- Users should be able to upload images from their local devices or integrate with cloud storage services.

- The system should support popular image formats such as JPEG, PNG, and GIF.

b. Image Resizing:

- Users should have the option to resize images by specifying dimensions, aspect ratios, or using preset resizing options.

- The system should perform resizing operations while maintaining image quality and aspect ratio.

c. Batch Resize:

- Users should be able to select and resize multiple images simultaneously in a batch operation.

- The system should efficiently handle and process multiple image resize requests.

d. Image Download:

- Users should be able to download the resized images in their preferred format (e.g., JPEG, PNG).

- Provide options for individual or bulk download of the resized images.

e. User Management:

- Implement user authentication and account management functionalities.

- Users should be able to create accounts, log in, and manage their uploaded images and preferences.

f. Image Editing:

- Provide basic image editing features such as cropping, rotating, and adjusting brightness/contrast.

- Users should be able to apply filters or effects to enhance or modify the images.

**2. Non-Functional Requirements:**

a. Performance:

- The system should be optimized for fast and responsive image resizing, even with large file sizes.

- Minimize latency and provide quick feedback to users during the resizing process.

b. Scalability:

- Design the system to handle a high volume of concurrent image resize requests.

- Implement horizontal scaling techniques to accommodate increasing user demand.

c. Security:

- Ensure secure image upload and storage, protecting user data and privacy.

- Implement appropriate measures to prevent unauthorized access or malicious activities.

d. Usability:

- Design an intuitive and user-friendly interface with clear instructions and visual cues.

- Support responsive design to provide a consistent experience across different devices and screen sizes.

e. Error Handling and Logging:

- Handle and display appropriate error messages when image upload or resizing fails.

- Implement logging mechanisms to track and monitor errors for troubleshooting and system improvement.

f. Integration:

- Integrate with popular cloud storage services (e.g., Google Drive, Dropbox) for seamless image upload and retrieval.

- Provide APIs or integration options for developers to use the image resize functionality in their own applications.

**TECHNICAL ARCHITECTURE**

Technical Architecture for Image Resize Website:

The technical architecture of an image resize website typically involves several components and layers working together to provide the desired functionality. Here's a high-level overview of the technical architecture:

1. User Interface (UI) Layer:

- The UI layer provides the user interface through which users interact with the website.

- It includes the web pages, forms, buttons, and other elements that allow users to upload images, specify resizing options, and download the resized images.

- HTML, CSS, and JavaScript are commonly used technologies for implementing the UI layer.

2. Application Layer:

- The application layer handles the business logic and processing of user requests.

- It receives user input from the UI layer and orchestrates the necessary operations to resize images.

- This layer interacts with the image processing layer to perform the actual resizing of images.

- It may also involve additional functionalities such as image editing, validation, and user management.

3. Image Processing Layer:

- The image processing layer is responsible for performing the actual resizing and optimization of images.

- It receives the original images from the application layer and applies resizing algorithms to generate the resized images.

- This layer can leverage image processing libraries or frameworks to efficiently manipulate and optimize images.

- It should handle different image formats, maintain aspect ratios, and optimize the resized images while preserving quality.

4. Data Storage Layer:

- The data storage layer manages the storage and retrieval of user-uploaded images and resized images.

- It can utilize a combination of file storage systems and databases.

- Uploaded images can be stored temporarily for processing or permanently for user access and retrieval.

- Resized images can be stored in a cache for faster retrieval or persisted for future downloads.

5. Infrastructure Layer:

- The infrastructure layer includes the underlying infrastructure components needed to run the application.

- It may involve cloud services or on-premises infrastructure, depending on the deployment model.

- This layer encompasses servers, networking, storage, and other resources required for hosting and scaling the application.

- Infrastructure-as-a-Service (IaaS) or Platform-as-a-Service (PaaS) providers can be utilized for managing the infrastructure.

6. Integration and APIs:

- The image resize website may integrate with external services such as cloud storage providers, authentication systems, or image editing tools.

- APIs or SDKs can be used to facilitate integration and enable seamless interaction with these services.

- Integration with content delivery networks (CDNs) can also be considered for improved performance and global image delivery.

7. Security and Authentication:

- The technical architecture should incorporate appropriate security measures to protect user data, prevent unauthorized access, and ensure secure image uploads and downloads.

- User authentication mechanisms, such as OAuth or JWT, can be implemented to secure user accounts and access to the image resize functionality.

8. Monitoring and Analytics:

- Incorporate monitoring and analytics tools to track the performance, usage patterns, and errors within the application.

- These tools can provide insights into system health, user behavior, and help identify areas for optimization and improvement.

9. Deployment and Scalability:

- The architecture should support easy deployment and scaling of the application.

- Containerization technologies such as Docker and container orchestration platforms like Kubernetes can be used to package and deploy the application components.

- Horizontal scaling can be achieved by adding more instances of the application to handle increased user demand.

**FRAME WORKS**

There are several frameworks available that can be used to develop an image resize website. Here are some popular frameworks commonly used for web development:

1. Django:

- Django is a high-level Python web framework that follows the Model-View-Controller (MVC) architectural pattern.

- It provides a robust set of features for building web applications, including user authentication, database ORM, and URL routing.

- Django's scalability, security features, and built-in admin interface make it a popular choice for complex web applications.

2. Flask:

- Flask is a lightweight Python web framework that follows a microservices architecture.

- It provides a minimalistic approach to web development, allowing developers to have more control over the application structure.

- Flask is suitable for small to medium-sized applications and provides flexibility in terms of choosing libraries and components.

3. Ruby on Rails:

- Ruby on Rails, often referred to as Rails, is a popular web application framework written in Ruby.

- Rails follows the MVC architectural pattern and emphasizes convention over configuration, promoting rapid development.

- It includes features like automated testing, scaffolding, and database migrations, making it efficient for developing web applications.

4. Express.js:

- Express.js is a minimalist web application framework for Node.js, a JavaScript runtime environment.

- Express.js focuses on simplicity and flexibility, allowing developers to build lightweight and scalable web applications.

- It provides a robust routing system, middleware support, and various plugins to extend the functionality.

5. Laravel:

- Laravel is a PHP web application framework known for its elegant syntax and developer-friendly features.

- It follows the MVC pattern and includes features like routing, database ORM, caching, and security measures.

- Laravel offers a rich set of tools and libraries for rapid development and has a large and active community.

6. ASP.NET:

- ASP.NET is a web application framework developed by Microsoft and is primarily used for building enterprise-level applications.

- It supports multiple programming languages such as C# and VB.NET and follows the MVC pattern.

- ASP.NET provides a wide range of features like built-in authentication, authorization, caching, and scalability options.

**THIRD PARTYS API**

IBM Watson APIs:

* IBM Watson provides a suite of APIs for advanced image analysis, recognition, and understanding.
* IBM Visual Recognition API can be used to analyze and identify objects, faces, and scenes in images.
* IBM Watson Language Translator API can be utilized to translate text within images or provide multilingual support