

CLARIFY-IMAGE

**A PROJECT REPORT
For
Major Project (KCA451)
Session (2023-24)**

Submitted by

**PRASHANT GUPTA
(2200290140113)**

**Submitted in partial fulfilment of the
Requirements for the Degree of**

MASTER OF COMPUTER APPLICATION

**Under the Supervision of
Ms. Komal Salgotra
Assistant Professor**



Submitted to

**DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206**

(MARCH 2024)

CERTIFICATE

Certified that **Prashant Gupta 2200290140113** has carried out the project work having “**Clarify-image**” (**Major Project-KCA451**) for **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

Prashant Gupta (2200290140113)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Ms. Komal Salgotra
Assistant Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Dr. Arun Tripathi
Head
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

PRASHANT GUPTA

ABSTRACT

Clarify-image is a comprehensive project aimed at developing a robust and scalable media Streaming platform, closely resembling the popular image Streaming service Clarify image. This Clarify image project, undertaken with the goal of providing users with a familiar and feature-rich experience, encompasses the entire software development lifecycle from conceptualization to deployment. The project incorporates key functionalities of Clarify image, including user registration, profile creation, tweet posting, following and unfollowing, direct messaging, and real-time updates. The project incorporates key functionalities of Clarify image, including user registration, profile creation, tweet posting, following and unfollowing, direct messaging, and real-time updates. This Clarify image project, undertaken with the goal of providing users with a familiar and feature-rich experience, encompasses the entire software development lifecycle from conceptualization to deployment. Leveraging modern web technologies and frameworks, Clarify image is designed to be responsive, intuitive, and accessible across various devices. Key features of Clarify image include a user-friendly interface, a secure authentication system, seamless real-time communication through Web Socket technology, and efficient data storage utilizing a relational database. The project incorporates key functionalities of Clarify image, including user registration, profile creation, tweet posting, following and unfollowing, direct messaging, and real-time updates. The project also addresses considerations for user privacy and data security, implementing encryption and access control measures to safeguard user information.

ACKNOWLEDGEMENTS

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Ms. Komal Salgotra** for his guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions. Words are not enough to express my gratitude to **Dr. Arun Kumar Tripathi**, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help on various occasions. Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions. Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Prashant Gupta

TABLE OF CONTENTS

Certificate.....	I
Abstract.....	II
Acknowledgement.....	III
Table of Content.....	1-2
List of Abbreviation.....	3
List of figures.....	4
Chapter 1. Introduction.....	5-7
1.1 Overview.....	5
1.2 Project Description.....	5-6
1.2.1 Project Scope.....	5
1.2.2 Target Audience.....	6
1.3 Hardware / Software.....	6-7
1.3.1 Hardware Requirement.....	6
1.3.2 Software Requirement.....	7
Chapter 2. Feasibility Study.....	8-10
2.1 Technical Feasibility.....	8
2.2 Operational Feasibility.....	9
2.3 Behavioural Feasibility.....	10

Chapter 3. Design.....	11-17
3.1 Use case	11-13
3.2 Flow Chart.....	14-16
3.3 ER Diagam.....	17
Chapter 4. Project Screenshots.....	18-28
4.1 Home Page.....	18-19
4.2 Editing Page.....	20-22
4.3 Doodle Page.....	23-25
4.4 Brush Tools Page.....	26-27
4.5 Filter Page.....	28
Chapter 5. Project Code.....	29-47
Chapter 6. Testing.....	48-73
API Introduction.....	48-50
5.1 Testing Level.....	51-54
5.2 Testing Techniques.....	55-57
5.2.1 White Box Testing.....	58-62
5.2.2 Control Structure Testing.....	63-66
5.2.3 Usability Testing.....	67-70
5.2.4 Integration Testing.....	70-73
Chapter 7. Future Scope and Conclusion.....	74-79
6.1 Future Scope.....	74-76
6.2 Conclusion.....	77-79
Chapter 8. Bibliography.....	80

LIST OF ABBREVIATIONS

SR. No	Abbreviation	Definition
1	HTML	Hypertext Mark-up language
2	CSS	Cascading Style Sheets
3	JS	JavaScript Web Scripting
4	Bootstrap	CSS Framework
5	API	Application Program Interface

LIST OF FIGURES

Figure. No	Name of figure	page Number
1	Use Case Diagram	13
2	Home Page	19
3	Editing Page	21
4	Doodles Page	23
5	Brush Tool Page	25
6	Filter Page	27

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The Clarify image project, is a comprehensive endeavour to replicate the functionality and user experience of the renowned microblogging platform, Clarify image. Motivated by the desire to understand and implement key features of a modern social media platform, this project encompasses a range of technical aspects spanning front-end development, back-end infrastructure, and real-time communication.

1.2 PROJECT DESCRIPTION

In today's fast-paced world, instant communication and sharing of ideas are paramount. Social media platforms like Clarify image have become a dominant force in this landscape, enabling real-time of the popular Clarify image platform, offering a familiar yet innovative experience for microblogging enthusiasts.

1.2.1 PROJECT SCOPE:

Clarify image will encompass the core functionalities of Clarify image, including:

- User registration and profiles: Create accounts, edit profiles.
- Microblogging: Post short messages with text, images, and images.
- Interactions: Edit, save, Modification.
- Direct messaging: Send Content to other users.
- Notifications: Receive updates on mentions, and direct messages.

1.2.2 TARGET AUDIENCE:

Clarify image targets individuals and communities seeking a dynamic platform for:

- Sharing quick updates
- Connecting with friends and like-minded individuals.
- Engaging with user for Better Interaction with the Images.

1.3 HARDWARE/SOFTWARE

1.3.1 HARDWARE REQUIREMENTS

Hardware- : Processor i3 or above

Clock speed- : 3.0 GHz

RAM size- : 4 GB or above

Hard Disk capacity : 500 GB or above

1.3.2 SOFTWARE REQUIREMENTS

Operating System- : Windows 10 / 11

Browser- : Google chrome or any other

Application software- : Visual Studio Code

Technology : HTML5, CSS3, JS, BootStrap5

Server Required	: XAMPP
Language Required	: HTML, CSS, JAVASCRIPT
Documentation	: MS Word

CHAPTER 2

FEASIBILITY STUDY

A feasibility study analyses the viability of a project to determine whether the project or venture is likely to succeed. The study is also designed to identify potential issues and problems that could arise while pursuing the project. A feasibility study evaluates a project's or system's practicality. As part of a feasibility study, the objective and rational analysis of a potential business or venture is conducted to determine its strengths and weaknesses, potential opportunities and threats, resources required to carry out, and ultimate success prospects. Two criteria should be considered when judging feasibility: the required cost and expected value. A feasibility study is a comprehensive evaluation of a proposed project that evaluates all factors critical to its success in order to assess its likelihood of success. Business success can be defined primarily in terms of ROI, which is the amount of profits that will be generated by the project.

2.1 TECHNICAL FEASIBILITY

Technical feasibility study is concerned with specifying equipment and software that will successfully satisfy the user requirement; the technical needs of the system may vary considerably. The facility to produce outputs in a given time. Our project is a web based application which is based on client-server based application. In this application every page as output is render from server to client so it is necessary that the page should be rendered in time. For this I have avoided more and more code in the page-load event. A technical feasibility study reviews the technical resources available for your project. This study determines if you have the right equipment, enough equipment, and the right technical knowledge to complete your project objectives. For example, if your project plan proposes creating 50,000 products per month, but you can only produce 30,000 products per month in your factories, this project isn't technically feasible. This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is

Capable of converting the ideas into working systems. Technical feasibility also involves the evaluation of the hardware,

Software, and other technical requirements of the proposed system. As an exaggerated example, an organization wouldn't want to try to put Star Trek's transporters in their building—currently, this project is not technically feasible.

2.2 BEHAVIOURAL FEASIBILITY

Behavioural feasibility is a critical aspect to consider when developing a Clarify image project. This feasibility study assesses whether users and stakeholders are likely to accept and adapt to the new microblogging platform based on their behavioural patterns, preferences, and expectations. The literature on behavioural feasibility for social media platforms, including microblogging services, can provide valuable insights into user behaviour and acceptance. Behavioural feasibility for a Clarify image project involves a comprehensive analysis of user behaviour, preferences, and cultural considerations. Drawing upon existing literature on user behaviour in social media, UX design, feature acceptance, community building, adoption patterns, and user feedback can provide a solid foundation for developing a microblogging platform that aligns with user expectations and encourages widespread adoption.

2.3 OPERATIONAL FEASIBILITY

Operation feasibility is used to check whether the project is operationally feasible or not. Our project is mainly different from the other system because of its web-support feature. So the measure for operational feasibility is something different from other system. Generally the operational feasibility is related to organization aspects. The change determination is as such that early product were either a man or group of men or the jobs based manual but now a day with the advent of Internet technology. This assessment involves undertaking a study to analyse and determine whether—and how well—the organization's needs can be met by completing the project. Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development.

This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility—helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide.

CHAPTER 3

DESIGN

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer's goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirements have been specified and analysed, system design is the first of the three technical activities - design, code and test that is required to build and verify software. The importance can be stated with a single word "Quality". Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer's view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage.

During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design. System Design is the process of designing the architecture, components, and interfaces for a system so that it meets the end-user requirements.

System Design for tech interviews is something that can't be ignored! Almost every IT giant whether it be Facebook, Amazon, Google, Apple or any other ask various questions based on System Design concepts such as scalability, load-balancing, caching, etc. in the interview. This specifically designed System Design tutorial will help you to learn and master System Design concepts in the most efficient way from basics to advanced level.

3.1 USE CASE DIAGRAM

In the Unified Modelling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. A use case diagram doesn't go into a lot of detail—for example, don't expect it to model the order in which steps are performed. Instead, a proper use case diagram depicts a high-level overview of the relationship between use cases, actors, and systems. Experts recommend that use case diagrams be used to supplement a more descriptive textual use case.

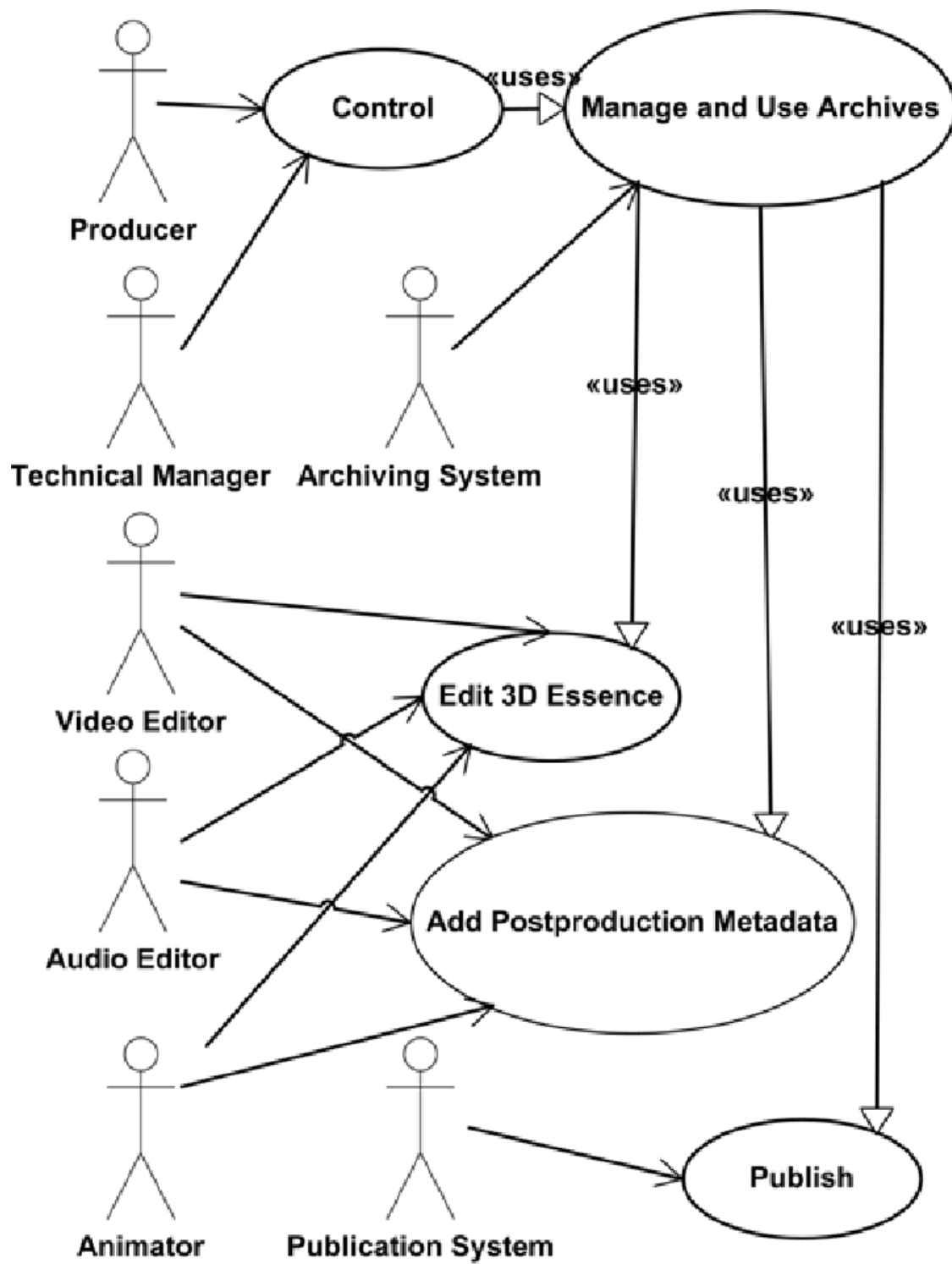
UML is the modelling toolkit that you can use to build your diagrams. Use cases are represented with a labelled oval shape. Stick figures represent actors in the process, and the actor's participation in the system is modelled with a line between the actor and use case. To depict the system boundary, draw a box around the use case itself.

These diagrams are used at a very high level of design. This high level design is refined again and again to get a complete and practical picture of the system. A well-structured use case also describes the pre-condition, post condition, and exceptions. These extra elements are used to make test cases when performing the testing.

Although use case is not a good candidate for forward and reverse engineering, still they are used in a slightly different way to make forward and reverse engineering. The same is true for reverse engineering. Use case diagram is used differently to make it suitable for reverse engineering.

Use case diagrams can be used for –

- Requirement analysis and high level design.
- Model the context of a system.
- Reverse engineering.
- Forward engineering.



3.2 FLOWCHART DIAGRAM

Flowcharts are nothing but the graphical representation of the data or the algorithm for a better understanding of the code visually. It displays step-by-step solutions to a problem, algorithm, or process. It is a pictorial way of representing steps that are preferred by most beginner-level programmers to understand algorithms of computer science, thus it contributes to troubleshooting the issues in the algorithm. A flowchart is a picture of boxes that indicates the process flow in a sequential manner. Since a flowchart is a pictorial representation of a process or algorithm, it's easy to interpret and understand the process. To draw a flowchart, certain rules need to be followed which are followed by all professionals to draw a flowchart and is widely accepted all over the countries.

Process flowchart: This type of flowchart shows all the activities that are involved in making a product. It basically provides a pathway to analyse the product to be built. A process flowchart is most commonly used in process engineering to illustrate the relation between the major as well as minor components present in the product. It is used in business product modelling to help understand employees about the project requirements and gain some insight about the project.

Data flowchart: As the name suggests, the data flowchart is used to analyse the data, specifically it helps in analysing the structural details related to the project. Using this flowchart, one can easily understand the data inflow and outflow from the system. It is most commonly used to manage data or to analyse information to and fro from the system.

Business Process Modelling Diagram: Using this flowchart or diagram, one can analytically represent the business process and help simplify the concepts needed to understand business activities and the flow of information. This flowchart illustrates the business process and models graphically which paves a way for process improvement.

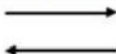
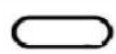



3.4 DATA FLOW DIAGRAM

DFD is the abbreviation for **Data Flow Diagram**. The flow of data of a system or a process is represented by DFD. It also gives insight into the inputs and outputs of each entity and the process itself. DFD does not have control flow and no loops or decision rules are present. Specific operations depending on the type of data can be explained by a flowchart. It is a graphical tool, useful for communicating with users, managers and other personnel. It is useful for analysing existing as well as proposed system.

It provides an overview of

- What data is system processes.
- What transformation are performed.
- What data are stored?
- What results are produced etc.?

Data Flow Diagram can be represented in several ways. The DFD belongs to structured-analysis modelling tools. Data Flow diagrams are very popular because they help us to visualize the major steps and data involved in software-system processes.

Symbol	Symbol Name	Description
	Flow Lines	Used to connect symbols
	Terminal	Used to start, pause or halt in the program logic
	Input/output	Represents the information entering or leaving the system
	Processing	Represents arithmetic and logical instructions
	Decision	Represents a decision to be made

The Data Flow Diagram has 4 components:

- **Process** Input to output transformation in a system takes place because of process function. The symbols of a process are rectangular with rounded corners, oval, rectangle or a circle. The process is named a short sentence, in one word or a phrase to express its essence

Data Flow Data flow describes the information transferring between different parts of the systems. The arrow symbol is the symbol of data flow.

- A relatable name should be given to the flow to determine the information which is being moved. Data flow also represents material along with information that is being Moved. Material shifts are modelled in systems that are not merely informative. A given flow should only transfer a single type of information. The direction of flow is represented by the arrow which can also be bi-directional.
- **Warehouse** The data is stored in the warehouse for later use. Two horizontal lines represent the symbol of the store. The warehouse is simply not restricted to being a data file rather it can be anything like a folder with documents, an optical disc, a filing cabinet. The data warehouse can be viewed independent of its implementation. When the data flow from the warehouse it is considered as data reading and when data flows to the warehouse it is called data entry or data updating.
- **Terminator** The Terminator is an external entity that stands outside of the system and communicates with the system. It can be, for example, organizations like banks, groups of people like customers or different departments of the same organization, which is not a part of the model system and is an external entity. Modelled systems also communicate with terminator.
- A relatable name should be given to the flow to determine the information which is being moved. Data flow also represents material along with information that is being Moved. Material shifts are modelled in systems that are not merely informative. A given flow should only transfer a single type of information. The direction of flow is represented by the arrow which can also be bi-directional
- **Warehouse** The data is stored in the warehouse for later use. Two horizontal lines represent the symbol of the store. The warehouse is simply not restricted to being a

Data file rather it can be anything like a folder with documents, an optical disc, a filing cabinet. The data warehouse can be viewed independent of its implementation.

When the data flow from the warehouse it is considered as data reading and when data flows to the warehouse it is called data entry or data updating.

- **Terminator** The Terminator is an external entity that stands outside of the system and communicates with the system. It can be, for example, organizations like banks, Groups of people like customers or different departments of the same organization, which is not a part of the model system and is an external entity. Modelled systems also communicate with terminator.

CHAPTER 4

PROJECT SCREENSHOTS

4.1 HOME PAGE

This module seamlessly takes care of user interaction with the Application. It enables the search on availability and manages the user search as per the search query. The home dashboard for image editing software typically includes a user-friendly interface that allows users to easily access various editing tools and features. Here's a brief overview of the home dashboard for some popular image editing software:

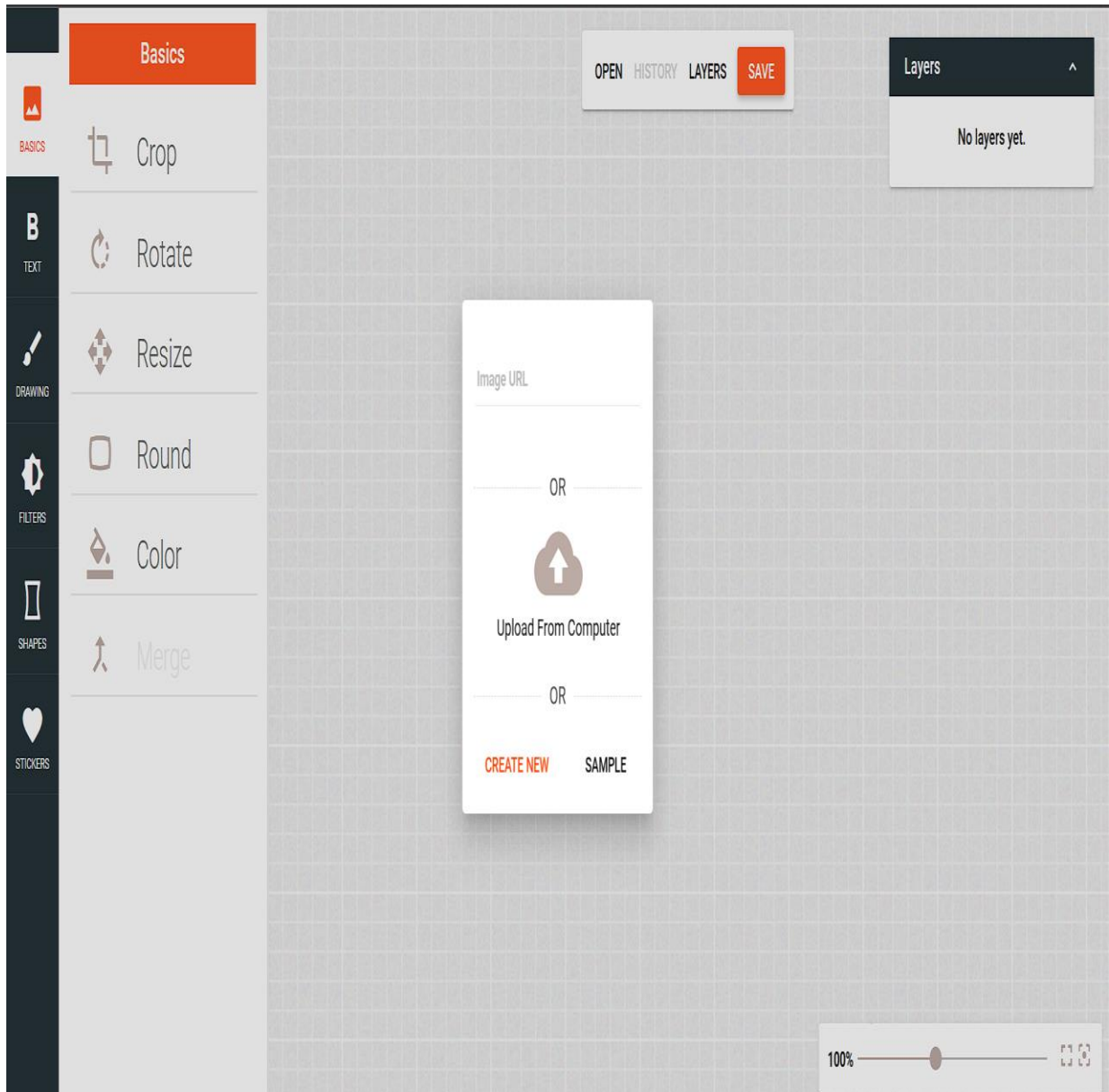
Canva: Offers a free online photo editor with a simple and intuitive interface, allowing users to edit images easily.

Picsart: Provides a comprehensive dashboard with a wide range of editing and design tools, catering to both beginners and professionals.

Pixlr: Features a suite of creative photo and design editing tools, including AI-powered generative tools, directly in the web browser or on mobile and desktop devices.

Img.vision: Offers an image editor add-on that allows users to edit images directly in the dashboard, with features like cropping, filtering, adding watermarks, and redacting faces.

WebFX: Recommends various image editing software, including Adobe Lightroom,



4.2 EDITING PAGE

This module seamlessly takes care of user interaction with the Application. This will help the user to visualize the content for editing Purpose.

The editing panel in image editing software typically includes a set of tools and features that allow users to manipulate and enhance digital images. This panel is the main interface where users can access various editing functions, such as:

Cropping, resizing, and rotating the image

Adjusting brightness, contrast, saturation, and other color settings

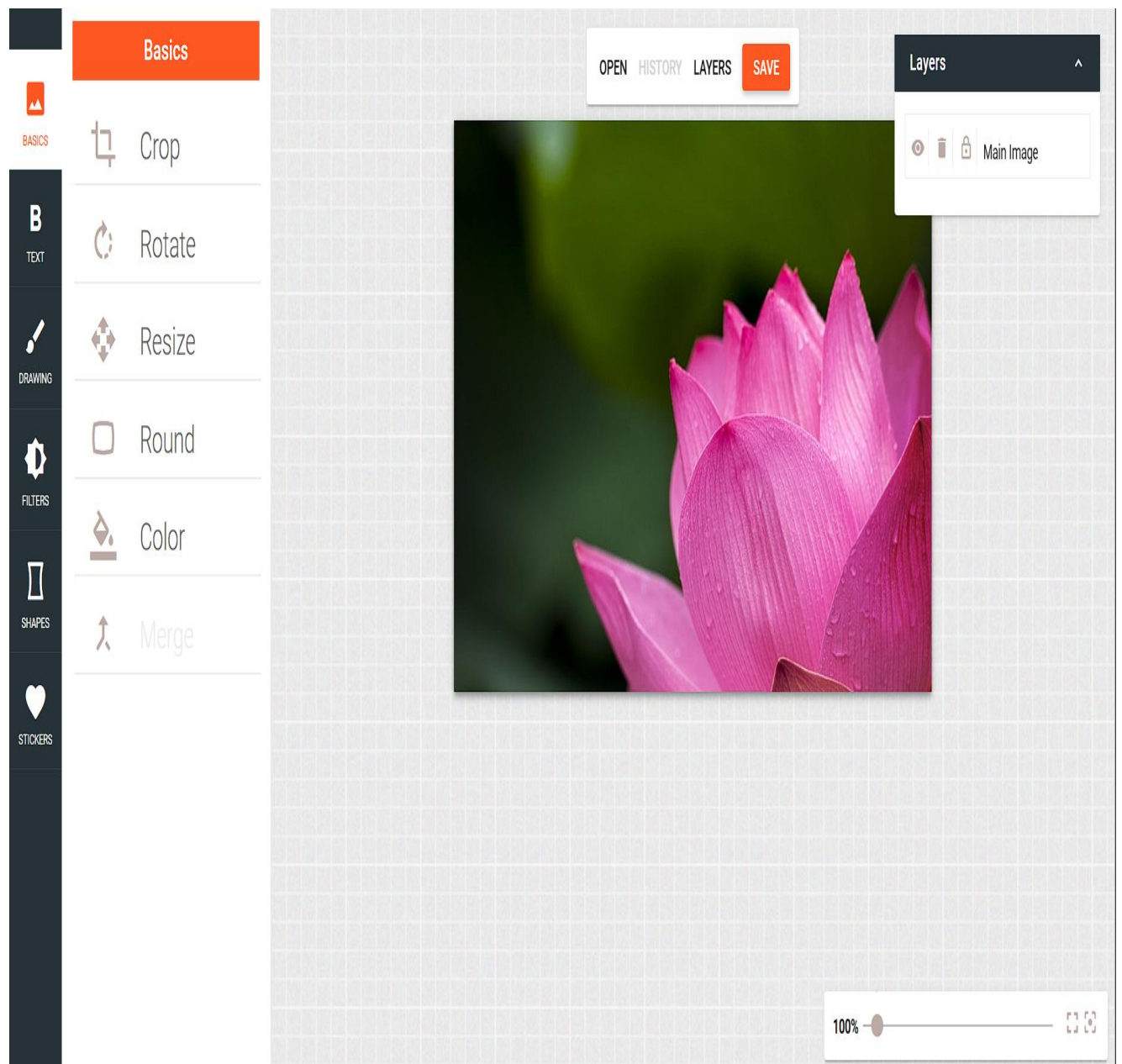
Applying filters, effects, and overlays to change the look and feel of the image

Adding text, shapes, lines, and other graphic elements

Retouching and enhancing specific areas of the image, like skin smoothing or eye enhancement

Masking and layering to isolate and work on different parts of the image independently

The editing panel is designed to be intuitive and user-friendly, with tools organized into logical sections for easy access. Many advanced editing panels also incorporate AI-powered features for faster and more accurate edits, like automatic face and body recognition.



4.3 DOODLES PAGE

It enables the search on availability and manages the user search as per the search query. The icons in the panel for image editing software typically include a set of tools and features that allow users to manipulate and enhance digital images. This panel is the main interface where users can access various editing functions, such as:

Cropping, resizing, and rotating the image

Adjusting brightness, contrast, saturation, and other color settings

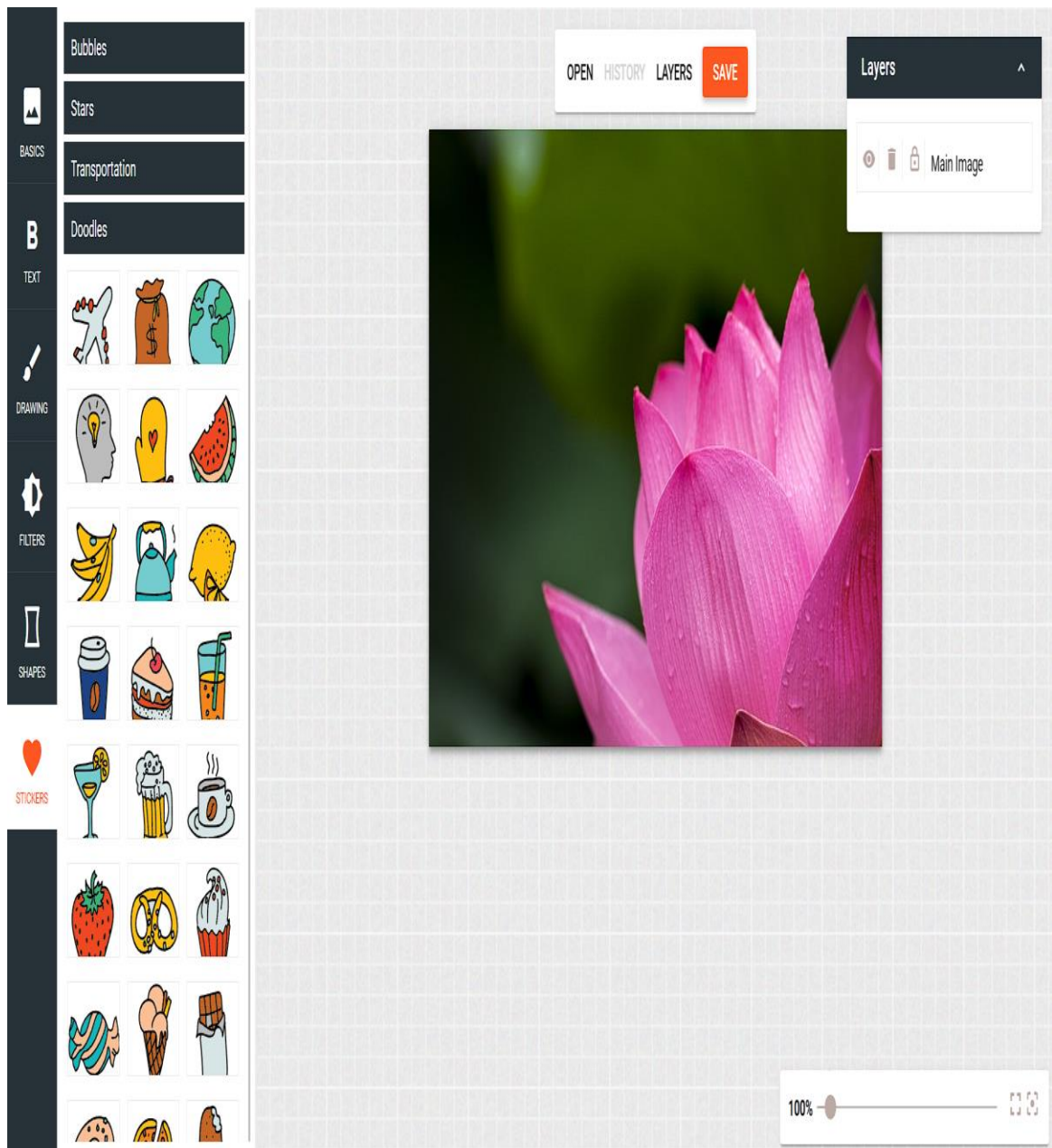
Applying filters, effects, and overlays to change the look and feel of the image

Adding text, shapes, lines, and other graphic elements

Retouching and enhancing specific areas of the image, like skin smoothing or eye enhancement

Masking and layering to isolate and work on different parts of the image independently

The icons in the editing panel are designed to be intuitive and user-friendly, with tools organized into logical sections for easy access. Many advanced editing panels also incorporate AI-powered features for faster and more accurate edits, like automatic face and body recognition.



4.4 BRUSH TOOLS PAGE

This module seamlessly takes care of user edit the image with the help of different kind of brush tools and modify the image with the help of these tools. The Brushes panel in image editing software provides access to various brush presets and settings that allow users to customize their painting experience. Here's a brief overview of the Brushes panel:

The Brushes panel displays a collection of preset brushes that users can select and apply to their images. These presets include different brush shapes, sizes, and textures.

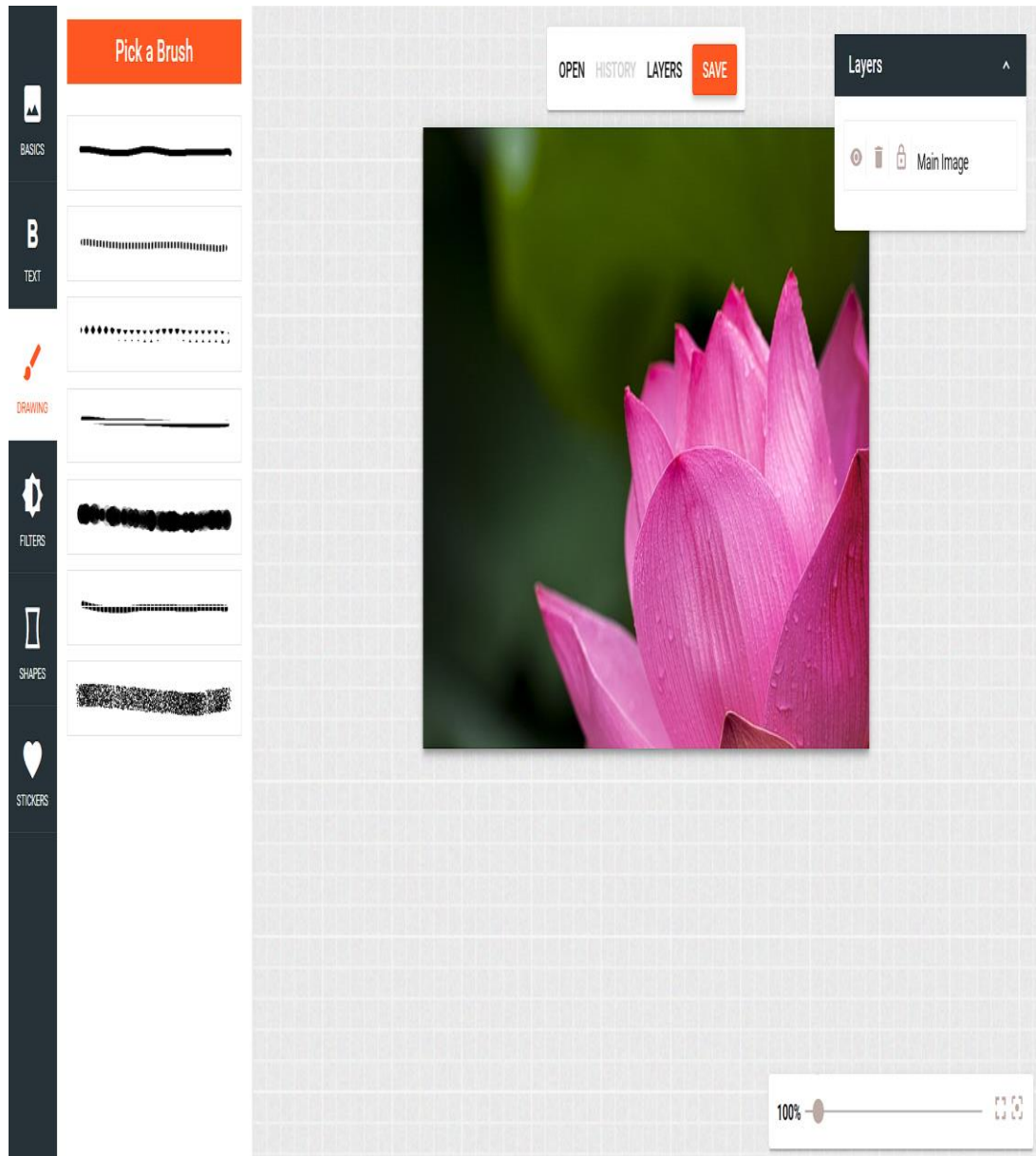
Users can modify the appearance and behavior of brushes by adjusting settings in the Brush Settings panel, such as brush tip options, brush dynamics, and texture. These settings determine how the paint is applied to the image.

Many image editing software allow users to create and save custom brushes by selecting a brush tip shape or using part of an image as a brush tip. This enables users to achieve unique painting effects.

Some software, like Photoshop, provide an extensive library of advanced brushes designed for specific purposes, such as sketching, inking, or painting with various media like oils or watercolors.

The Brushes panel is typically organized into categories or folders for easy navigation and selection of the desired brush preset.

Users can quickly access frequently used brush presets by saving them as favorites or creating custom brush sets.



4.5 FILTER PAGE

This module seamlessly takes care of user change the color opacity and apply the different type of filters on the image.

This will help the user to edit the image as per the colour scale. Filter is the main aspect in all kind of images, which adds the additional contrast factor in the image.

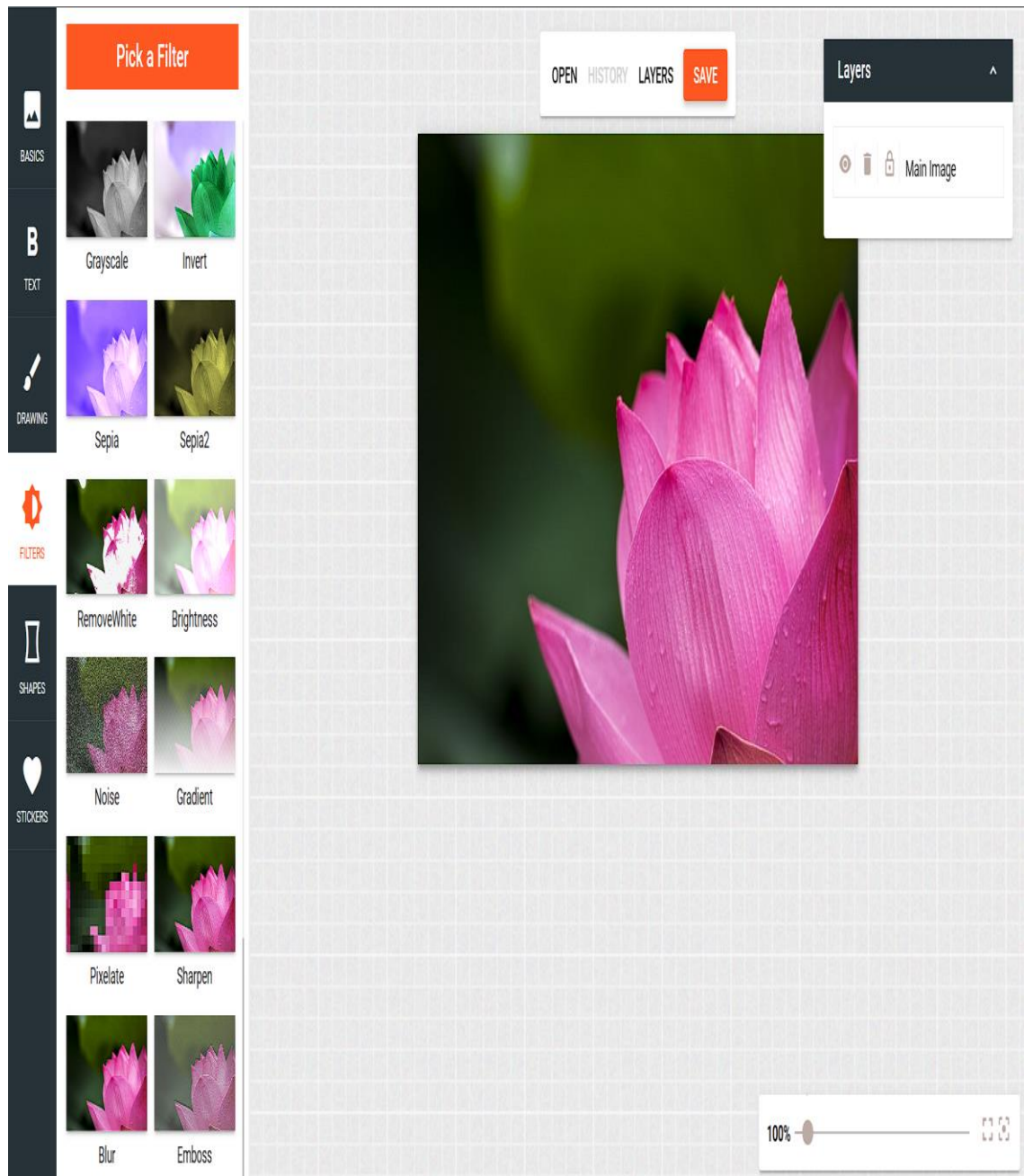
The Filter Panel in image editing software provides a collection of effects and filters that can be applied to images to enhance their appearance. Here's a brief overview of the Filter Panel in various software:

Image Tricks: Offers over 110 Core Image filters, 35 Quartz Compositions, and 70 image masks to enhance photos.

Filter Forge: Features a library of 6747 photo effects, including cartoon and watercolor effects, frames, distortions, and glitches, with the ability to generate seamless textures and create custom filters using a node-based editor.

Canva: Provides free online photo editing tools with creative photo effects to spice up images.

The Filter Panel is designed to simplify the process of applying effects and filters to images, often with customizable parameters to achieve desired results. These filters can transform images into unique and artistic creations, making them a valuable tool for photographers, digital artists, and anyone looking to enhance their photos.



CHAPTER 5

PROJECT CODE

Main File

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1" />
    <title>Image Editor</title>
    <style type="text/css">[ng-cloak]#splash{display:block!important}[ng-
cloak]{display:none}#splash{display:none;position:absolute;top:45%;left:50%;width:6em;height
:6em;overflow:hidden;border-radius:100%;z-index:0}@-webkit-keyframes
fade{from{opacity:1}to{opacity:.2}}@keyframes fade{from{opacity:1}to{opacity:.2}}@-
webkit-keyframes rotate{from{-webkit-transform:rotate(0deg)}to{-webkit-
transform:rotate(360deg)}}@keyframes
rotate{from{transform:rotate(0deg)}to{transform:rotate(360deg)}}#splash::after,#splash::before
{content:";position:absolute;top:0;left:0;width:100%;height:100%}#splash::before{background:l
inear-gradient(to right,green,#ff0);-webkit-animation:rotate 2.5s linear infinite;animation:rotate
2.5s linear infinite}#splash::after{background:linear-gradient(to bottom,red,#00f);-webkit-
animation:fade 2s infinite alternate,rotate 2.5s linear reverse infinite;animation:fade 2s infinite
alternate,rotate 2.5s linear reverse infinite}#splash-
spinner{position:absolute;width:100%;height:100%;z-index:1;border-radius:100%;box-
sizing:border-box;border-left:.5em solid transparent;border-right:.5em solid transparent;border-
bottom:.5em solid rgba(255,255,255,.3);border-top:.5em solid rgba(255,255,255,.3);-webkit-
animation:rotate .8s linear infinite;animation:rotate .8s linear infinite}</style>
    <link rel="stylesheet" href="assets/css/main.css?v23">
```



```

    <link href='http://fonts.googleapis.com/css?family=Roboto:300,400,500,700,900'
rel='stylesheet' type='text/css'>
</head>

<body id="editor" ng-app="ImageEditor" ng-strict-di>

    <div id="splash" ng-cloak>
        <div id="splash-spinner"></div>
    </div>

    <div ng-controller="MainController" ng-cloak style="height: 100%">
        <section id="right-panels">
            <section id="objects" class="md-whiteframe-z1" ng-
controller="ObjectsPanelController">
                <h3 ed-toggle-panel-visibility="objects"><span>Layers</span><i class="mdi mdi-
expand-less"></i></h3>
                <div class="scrollbar-wrapper" ed-pretty-scrollbar ed-scroll-axis="y">
                    <ul class="objects-panel list-unstyled" ui-sortable="sortableOptions" ng-
model="objects">
                        <li class="object panel-item" ng-repeat="object in objects" ng-
click="setAsActive(object)" ng-if="!object.ignore">
                            <i class="mdi mdi-remove-red-eye toggle-visibility" ng-class="{ 'not-
visible': !object.visible }" ng-click="toggleVisibility(object); $event.stopPropagation();"></i>
                            <i class="mdi mdi-delete delete-object" ng-click="deleteObject(object);
$event.stopPropagation();"></i>
                            <i class="mdi mdi-lock-outline lock-object" ng-class="{ 'not-locked':
!object.locked }" ng-click="toggleLock(object); $event.stopPropagation();"></i>
                            {{ specify(object.name) || specify(object.type) }}
                        </li>
                        <li ng-if="!canvas.fabric._objects.length" class="text-center"><h4>No layers
yet.</h4></li>
                    </ul>
                </div>
            </section>
        </div>
    </div>

```

</section>

```
<section id="history" class="md-whiteframe-z1 panel-hidden">
  <h3 ed-toggle-panel-visibility="history"><span>History</span><i class="mdi mdi-
expand-less"></i></h3>
  <ul class="history-panel list-unstyled" ed-pretty-scrollbar ed-scroll-axis="y">
    <li class="history-item panel-item" ng-click="history.load(item)" ng-repeat="item
in history.all">
      <i class="mdi mdi-{{ item.icon }} main-icon"></i>
      {{ specify(item.name) }}
    </li>
    <li ng-if="history.all.length === 0" class="text-center"><h4>No history items
yet.</h4></li>
  </ul>
</section>
</section>
```

```
<section ng-hide="!loading" id="spinner">
  <md-progress-circular class="md-spinner" md-diameter="100" md-
mode="indeterminate"></md-progress-circular>
</section>
```

```
<section id="left-sidebar" ed-tabs="basics" class="md-whiteframe-z1">
  <div id="sidebar-toggler" ed-toggle-sidebar>
    <md-button class="md-fab md-primary" aria-label="Toggle Sidebar">
      <i class="mdi mdi-edit show-icon"></i>
      <i class="mdi mdi-close close-icon"></i>
    </md-button>
  </div>
  <div id="navigation-bar">
    <div data-activates="basics" class="nav-item">
      <i class="mdi mdi-image"></i>
      <div class="title">BASICS</div>
```

```

</div>
<div data-activates="text" class="nav-item">
  <i class="mdi mdi-format-bold"></i>
  <div class="title pull-up">TEXT</div>
</div>
<div data-activates="drawing" class="nav-item">
  <i class="mdi mdi-brush"></i>
  <div class="title">DRAWING</div>
</div>
<div data-activates="filters" class="nav-item">
  <i class="mdi mdi-brightness-6"></i>
  <div class="title">FILTERS</div>
</div>
<div data-activates="simple-shapes" class="nav-item">
  <i class="mdi mdi-panorama-vertical"></i>
  <div class="title">SHAPES</div>
</div>
<div data-activates="stickers" class="nav-item">
  <i class="mdi mdi-favorite"></i>
  <div class="title pull-up">STICKERS</div>
</div>
</div>
<div id="tabs">
  <section id="basics" data-index="1" class="tab">
    <h2 class="tab-title">Basics</h2>
    <div ng-controller="CropController" class="basics-item">
      <div ng-click="cropper.start($event)">
        <i class="mdi mdi-crop basics-icon"></i>
        <div class="icon-title">Crop</div>
      </div>

      <div data-name="crop" class="actions-menu md-whiteframe-z1" ng-
class="activePanel == 'crop' ? 'open' : 'closed'">

```

```

<div class="menu-header"><h3>Crop</h3></div>
<div class="row">
  <md-input-container class="col-sm-6">
    <label>Width</label>
    <input type="text" ng-model="width" ng-
change="cropzone.setWidth(width)">
  </md-input-container>
  <md-input-container class="col-sm-6">
    <label>Height</label>
    <input type="text" ng-model="height" ng-
change="cropzone.setHeight(height)">
  </md-input-container>
</div>
<div class="buttons">
  <md-button ng-click="cropper.stop()">cancel</md-button>
  <md-button ng-click="cropper.crop()" class="md-primary md-
raised">crop</md-button>
</div>
</div>
</div>

<div ng-controller="RotateController" class="basics-item">
  <div ng-click="startRotator($event)">
    <i class="mdi mdi-rotate-right basics-icon"></i>
    <div class="icon-title">Rotate</div>
  </div>

  <div data-name="rotate" class="actions-menu md-whiteframe-z1" ng-
class="activePanel == 'rotate' ? 'open' : 'closed'">
    <div class="menu-header"><h3>Rotate</h3></div>

    <div class="rotate-icons">
      <i class="mdi mdi-rotate-left" ng-click="rotateProper(-90)"></i>

```

```

        <i class="mdi mdi-rotate-right" ng-click="rotateProper(90)"></i>
    </div>

    <section class="menu-group">
        <div class="md-slider">
            <div class="slider-label">Angle</div>
            <md-slider aria-label="Angle" step="1" ng-model="angle" ng-
change="rotate(angle)" min="0" max="360" ></md-slider>
        </div>
        <div class="slider-input">
            <input type="number" ng-model="angle" ng-change="rotate(angle)"
min="0" max="360"/>
        </div>
    </section>

    <div class="buttons">
        <md-button ng-click="cancel()">cancel</md-button>
        <md-button ng-click="applyRotation()" class="md-primary md-
raised">apply</md-button>
    </div>
</div>

<div ng-controller="ResizeController" class="basics-item">
    <div ng-click="startResizer($event)">
        <i class="mdi mdi-open-with basics-icon"></i>
        <div class="icon-title">Resize</div>
    </div>

    <div data-name="resize" class="actions-menu md-whiteframe-z1" ng-
class="activePanel == 'resize' ? 'open' : 'closed'">
        <div class="menu-header"><h3>Resize</h3></div>

```

```

<div class="clearfix">
  <md-input-container class="col-sm-6">
    <label>Width</label>
    <input type="number" ng-model="width" ng-
change="aspectToHeight(width)">
  </md-input-container>

  <md-input-container class="col-sm-6">
    <label>Height</label>
    <input type="number" ng-model="height" ng-
change="aspectToWidth(height)">
  </md-input-container>
</div>

<md-checkbox ng-model="usePercentages" ng-
change="togglePercentages(usePercentages)">Use Percentages</md-checkbox>
<md-checkbox ng-model="constrainProportions">Maintain Aspect
Ratio</md-checkbox>

<div class="buttons margin-top">
  <md-button ng-click="close()">close</md-button>
  <md-button ng-click="apply()" class="md-primary md-
raised">resize</md-button>
</div>
</div>
</div>

<div ng-controller="RoundedCornersController" class="basics-item">
  <div ng-click="startRoundedCorners($event)">
    <i class="mdi mdi-panorama-wide-angle basics-icon"></i>
    <div class="icon-title">Round</div>
  </div>

```

```

<div data-name="round" class="actions-menu md-whiteframe-z1" ng-
class="activePanel == 'round' ? 'open' : 'closed'">
  <div class="menu-header"><h3>Round</h3></div>

  <section class="menu-group">
    <div class="md-slider">
      <div class="slider-label">Radius</div>
      <md-slider aria-label="Radius" step="1" ng-model="radius" ng-
change="adjustPreview()" min="1" max="350" ></md-slider>
    </div>
    <div class="slider-input">
      <input type="number" ng-model="radius" ng-
change="adjustPreview()"/>
    </div>
  </section>

  <div class="buttons">
    <md-button ng-click="cancel()">Cancel</md-button>
    <md-button ng-click="apply()" class="md-primary md-
raised">Apply</md-button>
  </div>
</div>

<div ng-controller="CanvasBackgroundController" class="basics-item">
  <div ng-click="openPanel('canvasColor', $event)">
    <i class="mdi mdi-format-color-fill basics-icon"></i>
    <div class="icon-title">Color</div>
  </div>

  <div data-name="canvasColor" class="actions-menu md-whiteframe-z1 fluid"
ng-class="activePanel == 'canvasColor' ? 'open' : 'closed'">
    <div class="menu-header"><h3>Canvas Background Color</h3></div>

```

```

<input ed-flat ed-color-picker="setBackground(color)" type="text"/>

<div class="buttons">
  <md-button ng-click="cancel()">Cancel</md-button>
  <md-button ng-click="apply()" class="md-primary md-
raised">Apply</md-button>
</div>
</div>
</div>

<div class="basics-item" ng-class="{ disabled: canvas.fabric._objects.length <= 1
}">

  <div ng-click="canvas.mergeObjects()">
    <i class="mdi mdi-merge-type basics-icon"></i>
    <div class="icon-title">Merge</div>
  </div>
</div>
</section>

<section id="text" data-index="2" class="tab" ng-controller="TextController">
  <h2 class="tab-title">Pick a Font</h2>

  <div class="filter-inputs">
    <div class="filter-input">
      <md-select ng-model="filters.category" ng-
change="fonts.paginator.filter(filters)" placeholder="Select a Type">
        <md-option value="serif">Serif</md-option>
        <md-option value="sans-serif">Sans Serif</md-option>
        <md-option value="display">Display</md-option>
        <md-option value="handwriting">Handwriting</md-option>

```



```

        <md-option value="monospace">Monospace</md-option>
    </md-select>
</div>

<md-input-container class="filter-input">
    <label>Search</label>
    <input type="text" ng-model="filters.family" ng-
change="fonts.paginator.filter(filters)">
</md-input-container>
</div>

<div ng-hide="activeTab !== 'text'" class="pagination md-whiteframe-z1" ed-
fonts-pagination></div>

<section class="fonts-preview" ed-pretty-scrollbar ed-scroll-axis="y" ed-scroll-
theme="light">
    <div class="font" ng-style="{ 'font-family': font.family }" ng-repeat="font in
fonts.paginator.currentItems" ng-click="changeFont(font.family, $event)">{{ font.family
}}</div>
</section>

<div data-name="text" class="actions-menu wider md-whiteframe-z1" ng-
class="isPanelEnabled() ? 'open' : 'closed'">
    <div class="menu-header"><h3>Text</h3></div>

    <section class="actions-menu-inner" ng-style="{ 'max-height':
maxPanelHeight }" ed-pretty-scrollbar ed-scroll-axis="y" ed-scroll-theme="light">
        <div class="md-slider">
            <div class="slider-label">Opacity</div>
            <md-slider aria-label="text opacity" step="0.1" ng-model="opacity" ng-
change="text.setProperty('opacity', opacity)" min="0" max="1"></md-slider>
        </div>
        <div class="slider-input">

```

```

        <input min="0.1" max="1" step="0.1" type="number" ng-
change="text.setProperty('opacity', opacity)" ng-model="opacity"/>
    </div>

    <div class="md-slider">
        <div class="slider-label">Font Size</div>
        <md-slider aria-label="text size" step="1" ng-
change="text.setProperty('fontSize', fontSize)" ng-model="fontSize" min="1"
max="100"></md-slider>
    </div>
    <div class="slider-input">
        <input type="number" ng-change="text.setProperty('fontSize', fontSize)"
ng-model="fontSize"/>
    </div>

    <div class="text-styles">
        <div class="text-style-toolbar" ed-text-align-buttons>
            <i data-align="left" class="mdi mdi-format-align-left"></i>
            <i data-align="center" class="mdi mdi-format-align-center active"></i>
            <i data-align="right" class="mdi mdi-format-align-right"></i>
        </div>
        <div class="text-style-toolbar text-decoration-buttons" ed-text-decoration-
buttons>
            <div data-decoration="underline" class="underline toolbar-
btn">U</div>
            <div data-decoration="line-through" class="line-through toolbar-
btn">S</div>
            <div data-decoration="overline" class="overline toolbar-btn">O</div>
            <div data-decoration="italic" class="italic toolbar-btn">I</div>
        </div>
    </div>

    <div class="colorpicker">

```

```

    <div class="colorpicker-label"><span>Color</span></div>
    <input ed-color-picker="text.setProperty('fill', color)" type="text"/>
  </div>

```

```

<div class="colorpicker">
  <md-checkbox class="md-checkbox" ng-model="enableBackground" ng-
change="text.toggleProperty('backgroundColor', enableBackground)" aria-
label="Background">Background</md-checkbox>
  <input ed-color-picker="text.setProperty('backgroundColor', color)" ed-
color-picker-disabled="{{ !enableBackground }}" type="text"/>
</div>

```

```

<div class="colorpicker">
  <md-checkbox class="md-checkbox" ng-model="enableOutline" ng-
change="text.toggleProperty('stroke', enableOutline)" aria-label="Outline">Outline</md-
checkbox>
  <input ed-color-picker-disabled="{{ !enableOutline }}" ed-color-
picker="text.setProperty('stroke', color)" type="text"/>
</div>

```

```

<div class="md-inputs">
  <md-input-container>
    <label>Line Height</label>
    <input type="number" min="1" max="100" ng-model="lineHeight" ng-
change="text.setProperty('lineHeight', lineHeight)">
  </md-input-container>
  <md-input-container>
    <label>Outline Width</label>
    <input ng-disabled="!enableOutline" type="number" min="1"
max="100" ng-model="strokeWidth" ng-change="text.setProperty('strokeWidth',
strokeWidth)">
  </md-input-container>

```

```

</div>

<div class="buttons">
    <md-button ng-click="finishAddingTextToCanvas()" class="md-primary
md-raised">Save</md-button>
    <md-button ng-click="cancelAddingTextToCanvas()" class="md-
raised">Cancel</md-button>
</div>
</section>
</div>
</section>
<section id="drawing" class="tab" data-index="3" ng-
controller="DrawingController">
    <h2 class="tab-title">Pick a Brush</h2>
    <div id="brushes">
        <div ng-repeat="brush in drawing.availableBrushes" ng-class="{ active:
drawing.activeBrushName == brush }" class="brush" ng-click="changeBrush(brush, $event)">
            
        </div>
    </div>

    <div data-name="drawing" class="actions-menu wider md-whiteframe-z1" ng-
class="activePanel == 'drawing' ? 'open' : 'closed'">
        <div class="menu-header"><h3>Brush</h3></div>

        <section class="actions-menu-inner" ng-style="{ 'max-height':
maxPanelHeight }" ed-pretty-scrollbar ed-scroll-axis="y" ed-scroll-theme="light">
            <div class="md-slider">
                <div class="slider-label">Brush Thickness</div>
                <md-slider aria-label="brush thickness" step="1" ng-
model="drawing.params.brushWidth" ng-change="drawing.setProperty('width',
drawing.params.brushWidth)" min="1" max="100"></md-slider>

```

```

</div>
<div class="slider-input">
    <input type="number" ng-change="drawing.setProperty('width',
drawing.params.brushWidth)" ng-model="drawing.params.brushWidth"/>
</div>

<div class="colorpicker">
    <div class="colorpicker-label"><span>Color</span></div>
    <input ng-model="drawing.params.brushColor" ed-color-
picker="drawing.setProperty('color', color)" type="text"/>
</div>

<div class="heading-with-switcher">
    <h3>Shadow</h3>
    <md-switch class="md-switch" ng-
model="drawing.params.enableShadow" ng-
change="drawing.toggleShadow(drawing.params.enableShadow)" aria-label="Enable
Shadow"></md-switch>
</div>

<div class="md-slider">
    <div class="slider-label"><span>Blur</span></div>
    <md-slider aria-label="shadow blur" ng-
disabled="!drawing.params.enableShadow" step="1" ng-model="drawing.params.shadowBlur"
ng-change="drawing.setShadowProperty('blur', drawing.params.shadowBlur)" min="1"
max="100"></md-slider>
</div>
<div class="slider-input">
    <input type="number" ng-change="drawing.setShadowProperty('blur',
drawing.params.shadowBlur)" ng-disabled="!drawing.params.enableShadow" ng-
model="drawing.params.shadowBlur"/>
</div>

```

```

<div class="md-slider">
  <div class="slider-label"><span>Offset X</span></div>
  <md-slider ng-disabled="!drawing.params.enableShadow" aria-
label="shadow blur" step="1" ng-model="drawing.params.shadowOffsetX" ng-
change="drawing.setShadowProperty('offsetX', drawing.params.shadowOffsetX)" min="1"
max="100"></md-slider>
</div>
<div class="slider-input">
  <input type="number" ng-change="drawing.setShadowProperty('offsetX',
drawing.params.shadowOffsetX)" ng-disabled="!drawing.params.enableShadow" min="1" ng-
model="drawing.params.shadowOffsetX"/>
</div>

<div class="md-slider">
  <div class="slider-label"><span>Offset Y</span></div>
  <md-slider ng-disabled="!drawing.params.enableShadow" aria-
label="shadow blur" step="1" ng-model="drawing.params.shadowOffsetY" ng-
change="drawing.setShadowProperty('offsetY', drawing.params.shadowOffsetY)" min="1"
max="100"></md-slider>
</div>
<div class="slider-input">
  <input type="number" ng-change="drawing.setShadowProperty('offsetY',
drawing.params.shadowOffsetY)" ng-disabled="!drawing.params.enableShadow" min="1" ng-
model="drawing.params.shadowOffsetY"/>
</div>

<div class="colorpicker">
  <div class="colorpicker-label"><span>Color</span></div>
  <input ng-disabled="!drawing.params.enableShadow" ng-
model="drawing.params.shadowColor" start-color="{{ drawing.params.shadowColor }}" ed-
color-picker="drawing.setShadowProperty('color', color)" type="text"/>
</div>

```

```

        <div class="buttons">
            <md-button ng-click="finishAddingDrawingsToCanvas()" class="md-
primary md-raised">Save</md-button>
            <md-button ng-click="cancelAddingDrawingsToCanvas()" class="md-
raised">Cancel</md-button>
        </div>
    </section>
</div>
</section>
<section id="filters" class="tab" data-index="4" ng-controller="FiltersController">
    <h2 class="tab-title">Pick a Filter</h2>
    <div class="filters-list row" ed-pretty-scrollbar ed-scroll-axis="y" ed-scroll-
theme="light">
        <figure ng-class="{ applied: filters.filterAlreadyApplied(filter.name) }" ng-
repeat="filter in filters.all" ng-click="filters.applyFilter(filter)" class="filter col-sm-6">
            <div class="filter-image">
                
                <div ng-class="{ 'remove-only': !filter.options }" class="remove-filter-
overlay">
                    <i class="mdi mdi-cancel"></i>
                    <i ng-if="filter.options" ng-click="filters.lastAppliedFilter = filter;
$event.stopPropagation();" class="mdi mdi-settings"></i>
                </div>
            </div>
            <figcaption>{{ filter.displayName || filter.name }}</figcaption>
        </figure>
    </div>

    <div class="actions-menu md-whiteframe-z1" ng-
class="filters.lastAppliedFilter.options ? 'open' : 'closed'">

```

```

<div class="menu-header"><h3>{{ filters.lastAppliedFilter.name
}}</h3></div>

<section class="actions-menu-inner" ng-style="{ 'max-height':
maxPanelHeight }" ed-pretty-scrollbar ed-scroll-axis="y" ed-scroll-theme="light">
  <div ng-repeat="(key,value) in filters.lastAppliedFilter.options">

    <section ng-if="!value.colorpicker && !value.select">
      <div class="md-slider">
        <div class="slider-label">{{ key == filters.lastAppliedFilter.name ?
'amount' : key }}</div>
        <md-slider
          md-discrete
          aria-label="{{ key }}"
          step="{{ value.step ? value.step : 1 }}"
          ng-model="filters.lastAppliedFilter.options[key].current"
          ng-change="filters.applyValue(filters.lastAppliedFilter.name,
key, filters.lastAppliedFilter.options[key].current)"
          min="{{ value.min ? value.min : 1 }}"
          max="{{ value.max ? value.max : 250 }}"
        ></md-slider>
      </div>
      <div class="slider-input">
        <input ng-if="!value.colorpicker" type="number" ng-
change="filters.applyValue(filters.lastAppliedFilter.name, key,
filters.lastAppliedFilter.options[key].current)" ng-
model="filters.lastAppliedFilter.options[key].current"/>
      </div>
    </section>

    <section ng-if="value.colorpicker" class="colorpicker">
      <div class="colorpicker-label"><span>{{ key }}</span></div>

```



```

        <input ed-discrete ng-
model="filters.lastAppliedFilter.options[key].current" start-color="{ {
filters.lastAppliedFilter.options[key].current } }" ed-color-
picker="filters.applyValue(filters.lastAppliedFilter.name, key, color)" type="text"/>
    </section>

    <section ng-if="value.select">
        <div class="col-sm-8 pull-down">{{ key }}</div>
        <div class="col-sm-4 no-padding">
            <select class="form-control" ng-
model="filters.lastAppliedFilter.options[key].current" ng-
change="filters.applyValue(filters.lastAppliedFilter.name, key,
filters.lastAppliedFilter.options[key].current)">
                <option ng-repeat="mode in
filters.lastAppliedFilter.options[key].available" value="{{ mode }}">{{ mode }}</option>
            </select>
        </div>
    </section>
</div>
</section>
</div>
</section>

<section id="simple-shapes" class="tab" data-index="5" ng-
controller="SimpleShapesController">
    <h2 class="tab-title">Pick a Shape</h2>

</body>
</html>

```

CHAPTER 6

TESTING

Testing is a process of executing a program with the intent of finding bugs that makes the application fail to meet the expected behaviour. System Analysis and Design process including Requirement Analysis, Business Solution Options, Feasibility Study, and Architectural Design was discussed in previous chapter. Generally Software bugs will almost always exist in any software module. But it is not because of the carelessness or irresponsibility of programmer but because of the complexity. Humans have only limited ability to manage complexity. This chapter discusses about the testing of the solution and implementation methodologies. Regardless of the development methodology, the ultimate goal of testing is to make sure that what is created does what it is supposed to do. Testing plays a critical role for assuring quality and reliability of the software. I have included testing as a part of development process. The test cases should be designed with maximum possibilities of finding the errors or bugs. Software Testing is the process of executing a program or system with the intent of finding errors. The scope of software testing often includes examination of code as well as execution of that code in various environments and conditions. Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements. Testing stage of the project can be explained as below and system was tested for all these stages. Various level of testing are as follows

Why API?

API now became a new kind of business models and strategies for companies to equip in the big data era. Then what's so special? What makes API so hot these days? Let's briefly talk about what we can do with it in a business context.

APIs can generate massive amounts of value both internally and externally. Managing and processing data is one of the crucial factors in business management, and every company has built IT systems. As the size of the data is increasing exponentially, however, there is a limitation to cope with all the data through the traditional IT system. In this sense, applying API can be a solution with better efficiency and security. It can break down barriers between systems, which enables simplifying work processes, inter-cooperation between organizations and higher protection of data.

External merits of API are even fancier. If a company open their API either publicly or with extra fees, it can provide new services and acquire potential customers to their side. Customers can experience a higher level of services that weren't available before. By offering API services, third-party developers can build whole new kinds of products that even the companies had never thought of. For example, Google Map, the most popular API among developers, wasn't expected to draw that much effects at first. By Applying those data to real estate and various other fields, developers brought much higher values and assets back to Google.

Nowadays the number of API is continuously increasing, and this trend will be ongoing or even more. Now we can say that the implementation and management of API of a company is one of the critical factors for its competitive and strategic values.

The Clarify image Data API allows you to incorporate various Clarify image functions into your own website or application. You can use the API to perform operations such as uploading images, managing playlists and subscriptions, updating channel settings, and more

The API provides methods to retrieve, insert, update, and delete resources such as images, playlists, channels, and captions. It also supports features like searching for content, testing requests using the APIs Explorer, and calculating quota usage using the Quota Calculator

To get started with the Clarify image Data API, you need to obtain an API key, which can be done through the Google Cloud Console

. Once you have the API key, you can use it to authenticate your requests and access the API's functionalities. If you're new to APIs and need guidance on how to use the Clarify image API on your website, there are implementation guides and code samples available for various programming languages

To use the Clarify image Data API, you need to meet the following requirements:

API Key or OAuth 2.0 Token:

You need to specify an API key or provide an OAuth 2.0 token for every request.

The API key is available in the Developer Console's API Access pane of the project

Authorization:

Every insert, update, and delete request requires an authorization token

Some API methods for retrieving resources may support parameters that require authorization or may contain additional requirements

OAuth 2.0 Authentication:

The API supports the OAuth 2.0 authentication protocol, and you can provide an OAuth 2.0 token in the query parameter or using the HTTP authorization header

Quota and Limits:

The API has a daily request quota limit of 10,000 units. If you need more, you can request a Quota Extension to obtain an API key and enable the Clarify image Data API, you can follow the detailed guide provided by Google and other resources

Benefits of Using the Clarify image API

Many marketers and business owners are missing out on the benefits of incorporating Clarify image into their digital marketing strategy. Usually, it's a question of how much time it takes to create high-quality image content.

In a typical Clarify image marketing strategy, you would consistently publish image content every week with the hopes of generating likes and subscribers — a portion of which you'd hope would convert into customers. Luckily, this is no longer the only way to use Clarify image for your business. Clarify image has created an API that allows you to integrate your website with the social platform.

Here are some of the greatest benefits of using the Clarify image API:

You can take advantage of existing content. With the Clarify image API, you don't have to spend valuable time and resources creating content. Instead, you can take advantage of the content that has already been published to increase user engagement on your site.

Clarify image is one of the top social media platforms. With more than two billion active users in more than 100 countries, Clarify image is the top image streaming app. It's also the second-

Largest search engine, the second most popular social platform, and the second most visited site on the internet.

Featuring Clarify image's content can increase your dwell time. Because the Clarify image API can help you add Clarify image content to your website, it will help users stay on the page for longer as they watch the content that you've specifically curated for them. This increase in dwell

Time signals to search engines that your content is valuable, boosting your rankings for target keywords.

Using the API can help you convert more website visitors. Using the API is also valuable from a conversion rate and revenue standpoint. Prospects who engage with your content and stay on your website are more likely than not to make a purchase.

Using the Clarify image API is a no-brainer for many marketers, but Clarify image doesn't allow just anyone to access this valuable tool or its data. To prevent potential abuse, Clarify image grants each user a unique API key that allows them to connect their website or app to the platform.

Each API key is designated for an individual project. If you're using the API on two different websites, you should create two different API keys.

5.1 TESTING LEVELS

Software testing is a critical element of software quality assurance and represents the ultimate reuse of specification. Design and code testing represents interesting anomaly for the software during earlier definition and development phase, it was attempted to build software from an abstract concept to tangible implementation.

The testing phase involves, testing of the development of the system using various techniques such as White Box Testing, Control Structure Testing.

6.2 Software Strategies

A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level against customer requirements.

Unit testing: Unit testing tests the functionality of individual units of source code. It is the smallest component of a testable software that works in isolation with other parts of the code. I have done unit testing for various individual components of the source code to uncover errors within the boundary of the application.

Integration testing: Integration testing focuses on the design and construction of the software. Here the individual components that are tested using unit tests are combined and tested as a group. Its primary purpose is to expose the defects associated with the interfacing of modules. It checks if the modules perform the desired functionality when integrated together

System testing: System testing is performed on a completely integrated system to see if it meets the requirements, System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements. In system testing, integration testing passed components are taken as input.

Regression testing: Regression testing aims at verifying the functionality of the software that is previously tested and to which changes are made. It is to ensure the old software still works with new changes.

Acceptance testing: Acceptance testing is conducted to verify if the system compliance the business requirements. Software Testing is a method to check whether the actual

Software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements. Adhering to the levels of testing, Unit testing is performed on individual components of the system ensuring the expected behaviour. Later, I have integrated various components Together and performed Integration testing. Once the integration testing is done, I have performed System 30 testing and ensured the application works as per the requirements. Finally, acceptance testing is performed to check if the client accepts the system

- All its module are working properly and the project is best viewed both in Computer.
- The main requirement for this project that you should have good internet connection.

Performance Testing: Performance testing is performed to determine how well the system can perform in terms of responsiveness under all kinds of load. The web application is tested to see if it can sustain huge amount of requests providing higher throughput under different loads. I have simulated multiple hits on various pages of the application to evaluate the overall performance.

5.2 TESTING TECHNIQUES:

5.2.1 White Box Testing:

White box testing is a test case design method that uses the control structure of the procedural design to derive test cases. 88

5.2.2 Control Structure Testing

The following tests were conducted and it was noted that the BCBS is performing them well.

- Basic path Testing
- Condition Testing
- Data Flow Testing

- Loop Testing

There are many type of testing including in white box and black box testing.

5.2.3 Usability Testing:

- Usability testing is nothing but the User-friendliness check.
- In Usability testing, the application flow is tested so that a new user can understand the application easily.
- Basically, system navigation is checked in Usability testing.

Usability Test Cases:

- Web page content should be correct without any spelling or grammatical errors.
- All fonts should be same as per the requirements.
- All the text should be properly aligned.
- All the error mess

Ages should be correct without any spelling or grammatical errors and the error message should match with the field label.

- Tool tip text should be there for every field.
- All the fields should be properly aligned.
- Enough space should be provided between field labels, columns, rows, and error messages.
- All the buttons should be in a standard format and size.
- Home link should be there on every single page.
- Disabled fields should be greyed out. 89
- Check for broken links and images.
- Confirmation message should be displayed for any kind of update and delete operation.
- Check the site on different resolutions (640 x 480, 600x800 etc.?)
- Check the end user can run the system without frustration.
- Check the tab should work properly.
- Scroll bar should appear only if required.

- If there is an error message on submit, the information filled by the user should be there.
- Title should display on each web page

- All fields (Textbox, dropdown, radio button, etc.) and buttons should be accessible by keyboard shortcuts and the user should be able to perform all operations by using keyboard.
- Check if the dropdown data is not truncated due to the field size. Also, check whether the data is hardcoded or managed via administrator

5.2.4 Compatibility Testing

Compatibility testing is used to determine if your software is compatible with other elements of a system with which it should operate, e.g. Browsers, Operating Systems, or hardware.

Compatibility Test Scenarios:

- Test the website in different browsers (IE, Firefox, Chrome, Safari and Opera) and ensure the website is displaying properly.
- Test the HTML version being used is compatible with appropriate browser versions.
- Test the images display correctly in different browsers.
- Test the fonts are usable in different browsers.
- Test the java script code is usable in different browsers.
- Test the Animated GIF's across different browsers. 90

5.2.5 Integration Testing

In Database testing backend records are tested which have been inserted through the web or desktop applications.

The data which is displaying in the web application should match with the data stored in the Database.

Why API?

API now became a new kind of business models and strategies for companies to equip in the big data era. Then what's so special? What makes API so hot these days? Let's briefly talk about what we can do with it in a business context.

APIs can generate massive amounts of value both internally and externally. Managing and processing data is one of the crucial factors in business management, and every company has built IT systems. As the size of the data is increasing exponentially, however, there is a limitation to cope with all the data through the traditional IT system. In this sense, applying API can be a solution with better efficiency and security. It can break down barriers between systems, which enables simplifying work processes, inter-cooperation between organizations and higher protection of data.

External merits of API are even fancier. If a company open their API either publicly or with extra fees, it can provide new services and acquire potential customers to their side. Customers can experience a higher level of services that weren't available before. By offering API services, third-party developers can build whole new kinds of products that even the companies had never thought of. For example, Google Map, the most popular API among developers, wasn't expected to draw that much effects at first. By Applying those data to real estate and various other fields, developers brought much higher values and assets back to Google.

Nowadays the number of API is continuously increasing, and this trend will be ongoing or even more. Now we can say that the implementation and management of API of a company is one of the critical factors for its competitive and strategic values.

The Clarify image Data API allows you to incorporate various Clarify image functions into your own website or application. You can use the API to perform operations such as uploading images, managing playlists and subscriptions, updating channel settings, and more

The API provides methods to retrieve, insert, update, and delete resources such as images, playlists, channels, and captions. It also supports features like searching for content, testing requests using the APIs Explorer, and calculating quota usage using the Quota Calculator

To get started with the Clarify image Data API, you need to obtain an API key, which can be done through the Google Cloud Console

. Once you have the API key, you can use it to authenticate your requests and access the API's functionalities. If you're new to APIs and need guidance on how to use the Clarify image API on your website, there are implementation guides and code samples available for various programming languages

To use the Clarify image Data API, you need to meet the following requirements:

API Key or OAuth 2.0 Token:

You need to specify an API key or provide an OAuth 2.0 token for every request.

The API key is available in the Developer Console's API Access pane of the project

Authorization:

Every insert, update, and delete request requires an authorization token

Some API methods for retrieving resources may support parameters that require authorization or may contain additional requirements

OAuth 2.0 Authentication:

The API supports the OAuth 2.0 authentication protocol, and you can provide an OAuth 2.0 token in the query parameter or using the HTTP authorization header

Quota and Limits:

The API has a daily request quota limit of 10,000 units. If you need more, you can request a Quota Extension to obtain an API key and enable the Clarify image Data API, you can follow the detailed guide provided by Google and other resources

Benefits of Using the Clarify image API

Many marketers and business owners are missing out on the benefits of incorporating Clarify image into their digital marketing strategy. Usually, it's a question of how much time it takes to create high-quality image content.

In a typical Clarify image marketing strategy, you would consistently publish image content every week with the hopes of generating likes and subscribers — a portion of which you'd hope would convert into customers. Luckily, this is no longer the only way to use Clarify image for your business. Clarify image has created an API that allows you to integrate your website with the social platform.

Here are some of the greatest benefits of using the Clarify image API:

You can take advantage of existing content. With the Clarify image API, you don't have to spend valuable time and resources creating content. Instead, you can take advantage of the content that has already been published to increase user engagement on your site.

Clarify image is one of the top social media platforms. With more than two billion active users in more than 100 countries, Clarify image is the top image streaming app. It's also the second-

Largest search engine, the second most popular social platform, and the second most visited site on the internet.

Featuring Clarify image's content can increase your dwell time. Because the Clarify image API can help you add Clarify image content to your website, it will help users stay on the page for longer as they watch the content that you've specifically curated for them. This increase in dwell

Time signals to search engines that your content is valuable, boosting your rankings for target keywords.

Using the API can help you convert more website visitors. Using the API is also valuable from a conversion rate and revenue standpoint. Prospects who engage with your content and stay on your website are more likely than not to make a purchase.

Using the Clarify image API is a no-brainer for many marketers, but Clarify image doesn't allow just anyone to access this valuable tool or its data. To prevent potential abuse, Clarify image grants each user a unique API key that allows them to connect their website or app to the platform.

Each API key is designated for an individual project. If you're using the API on two different websites, you should create two different API keys.

5.1 TESTING LEVELS

Software testing is a critical element of software quality assurance and represents the ultimate reuse of specification. Design and code testing represents interesting anomaly for the software during earlier definition and development phase, it was attempted to build software from an abstract concept to tangible implementation.

The testing phase involves, testing of the development of the system using various techniques such as White Box Testing, Control Structure Testing.

6.2 Software Strategies

A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level against customer requirements.

Unit testing: Unit testing tests the functionality of individual units of source code. It is the smallest component of a testable software that works in isolation with other parts of the code. I have done unit testing for various individual components of the source code to uncover errors within the boundary of the application.

Integration testing: Integration testing focuses on the design and construction of the software. Here the individual components that are tested using unit tests are combined and tested as a group. Its primary purpose is to expose the defects associated with the interfacing of modules. It checks if the modules perform the desired functionality when integrated together

System testing: System testing is performed on a completely integrated system to see if it meets the requirements, System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements. In system testing, integration testing passed components are taken as input.

Regression testing: Regression testing aims at verifying the functionality of the software that is previously tested and to which changes are made. It is to ensure the old software still works with new changes.

Acceptance testing: Acceptance testing is conducted to verify if the system compliance the business requirements. Software Testing is a method to check whether the actual

Software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements. Adhering to the levels of testing, Unit testing is performed on individual components of the system ensuring the expected behaviour. Later, I have integrated various components Together and performed Integration testing. Once the integration testing is done, I have performed System 30 testing and ensured the application works as per the requirements. Finally, acceptance testing is performed to check if the client accepts the system

- All its module are working properly and the project is best viewed both in Computer.
- The main requirement for this project that you should have good internet connection.

Performance Testing: Performance testing is performed to determine how well the system can perform in terms of responsiveness under all kinds of load. The web application is tested to see if it can sustain huge amount of requests providing higher throughput under different loads. I have simulated multiple hits on various pages of the application to evaluate the overall performance.

5.2 TESTING TECHNIQUES:

5.2.1 White Box Testing:

White box testing is a test case design method that uses the control structure of the procedural design to derive test cases. 88

5.2.2 Control Structure Testing

The following tests were conducted and it was noted that the BCBS is performing them well.

- Basic path Testing
- Condition Testing
- Data Flow Testing
- Loop Testing

There are many type of testing including in white box and black box testing.

5.2.3 Usability Testing:

- Usability testing is nothing but the User-friendliness check.
- In Usability testing, the application flow is tested so that a new user can understand the application easily.
- Basically, system navigation is checked in Usability testing.

Usability Test Cases:

- Web page content should be correct without any spelling or grammatical errors.
- All fonts should be same as per the requirements.
- All the text should be properly aligned.
- All the error mess

Ages should be correct without any spelling or grammatical errors and the error message should match with the field label.

- Tool tip text should be there for every field.
- All the fields should be properly aligned.
- Enough space should be provided between field labels, columns, rows, and error messages.
- All the buttons should be in a standard format and size.
- Home link should be there on every single page.
- Disabled fields should be greyed out. 89
- Check for broken links and images.
- Confirmation message should be displayed for any kind of update and delete operation.
- Check the site on different resolutions (640 x 480, 600x800 etc.?)
- Check the end user can run the system without frustration.
- Check the tab should work properly.
- Scroll bar should appear only if required.
- If there is an error message on submit, the information filled by the user should be there.

- Title should display on each web page

- All fields (Textbox, dropdown, radio button, etc.) and buttons should be accessible by keyboard shortcuts and the user should be able to perform all operations by using keyboard.
- Check if the dropdown data is not truncated due to the field size. Also, check whether the data is hardcoded or managed via administrator

5.2.4 Compatibility Testing

Compatibility testing is used to determine if your software is compatible with other elements of a system with which it should operate, e.g. Browsers, Operating Systems, or hardware.

Compatibility Test Scenarios:

- Test the website in different browsers (IE, Firefox, Chrome, Safari and Opera) and ensure the website is displaying properly.
- Test the HTML version being used is compatible with appropriate browser versions.
- Test the images display correctly in different browsers.
- Test the fonts are usable in different browsers.
- Test the java script code is usable in different browsers.
- Test the Animated GIF's across different browsers. 90

5.2.5 Integration Testing

In Database testing backend records are tested which have been inserted through the web or desktop applications.

The data which is displaying in the web application should match with the data stored in the Database.

Test Cases for Integration Testing:

- Verify the database name: The database name should match with the specifications.

- Verify the Tables, columns, column types and defaults: All things should match with the specifications.

- Verify whether the column allows a null or not.

- Verify the Primary and foreign key of each table.

- Verify the Stored Procedure:

- Test whether the Stored procedure is installed or not.

- Verify the Stored procedure name

- Verify the parameter names, types and number of parameters.

- Test the parameters if they are required or not.

- Test the stored procedure by deleting some parameters

- Test when the output is zero, the zero records should be affected.

- Test the stored procedure by writing simple SQL queries.

- Test whether the stored procedure returns the values

- Test the stored procedure with sample input data.

- Verify the behaviour of each flag in the table.

- Verify the data gets properly saved into the database after each page submission.

- Verify the data if the DML (Update, delete and insert) operations are performed. 91

- Check the length of every field: The field length in the back end and front end must be same.

- Verify the database names of QA, UAT and production. The names should be unique.

- Verify the encrypted data in the database.

- Verify the database size. Also test the response time of each query executed.
- Verify the data displayed on the front end and make sure it is same in the back end.
- Verify the data validity by inserting the invalid data in the database.
- Verify the Triggers.

What is Security Testing?

Security Testing involves the test to identify any flaws and gaps from a security point of view.

Test Scenarios for Security Testing:

- Verify the web page which contains important data like password, credit card numbers, secret answers for security question etc should be submitted via HTTPS (SSL).
- Verify the important information like password, credit card numbers etc should display in encrypted format.
- Verify password rules are implemented on all authentication pages like Registration, forgot password, change password.
- Verify if the password is changed the user should not be able to login with the old password.
- Verify the error messages should not display any important information.
- Verify if the user is logged out from the system or user session was expired, the user should not be able to navigate the site.
- Verify to access the secured and non-secured web pages directly without login.
- Verify the "View Source code" option is disabled and should not be visible to the user. 92

- Verify the user account gets locked out if the user is entering the wrong password several times.
- Verify the cookies should not store passwords.
- Verify if, any functionality is not working, the system should not display any application, server, or database information. Instead, it should display the custom error page.
- Verify the SQL injection attacks.
- Verify the user roles and their rights. For Example, the requestor should not be able to access the admin page.
- Verify the important operations are written in log files, and that information should be traceable.
- Verify the session values are in an encrypted format in the address bar.
- Verify the cookie information is stored in encrypted format.
- Verify the application for Brute Force Attacks.

Why API?

API now became a new kind of business models and strategies for companies to equip in the big data era. Then what's so special? What makes API so hot these days? Let's briefly talk about what we can do with it in a business context.

APIs can generate massive amounts of value both internally and externally. Managing and processing data is one of the crucial factors in business management, and every company has built IT systems. As the size of the data is increasing exponentially, however, there is a limitation to cope with all the data through the traditional IT system. In this sense, applying API can be a solution with better efficiency and security. It can break down barriers between systems, which enables simplifying work processes, inter-cooperation between organizations and higher protection of data.

External merits of API are even fancier. If a company open their API either publicly or with extra fees, it can provide new services and acquire potential customers to their side. Customers can experience a higher level of services that weren't available before. By offering API services, third-party developers can build whole new kinds of products that even the companies had never thought of. For example, Google Map, the most popular API among developers, wasn't expected to draw that much effects at first. By Applying those data to real estate and various other fields, developers brought much higher values and assets back to Google.

Nowadays the number of API is continuously increasing, and this trend will be ongoing or even more. Now we can say that the implementation and management of API of a company is one of the critical factors for its competitive and strategic values.

The Clarify image Data API allows you to incorporate various Clarify image functions into your own website or application. You can use the API to perform operations such as uploading images, managing playlists and subscriptions, updating channel settings, and more

The API provides methods to retrieve, insert, update, and delete resources such as images, playlists, channels, and captions. It also supports features like searching for content, testing requests using the APIs Explorer, and calculating quota usage using the Quota Calculator

To get started with the Clarify image Data API, you need to obtain an API key, which can be done through the Google Cloud Console

. Once you have the API key, you can use it to authenticate your requests and access the API's functionalities. If you're new to APIs and need guidance on how to use the Clarify image API on your website, there are implementation guides and code samples available for various programming languages

To use the Clarify image Data API, you need to meet the following requirements:

API Key or OAuth 2.0 Token:

You need to specify an API key or provide an OAuth 2.0 token for every request.

The API key is available in the Developer Console's API Access pane of the project

Authorization:

Every insert, update, and delete request requires an authorization token

Some API methods for retrieving resources may support parameters that require authorization or may contain additional requirements

OAuth 2.0 Authentication:

The API supports the OAuth 2.0 authentication protocol, and you can provide an OAuth 2.0 token in the query parameter or using the HTTP authorization header

Quota and Limits:

The API has a daily request quota limit of 10,000 units. If you need more, you can request a Quota Extension to obtain an API key and enable the Clarify image Data API, you can follow the detailed guide provided by Google and other resources

Benefits of Using the Clarify image API

Many marketers and business owners are missing out on the benefits of incorporating Clarify image into their digital marketing strategy. Usually, it's a question of how much time it takes to create high-quality image content.

In a typical Clarify image marketing strategy, you would consistently publish image content every week with the hopes of generating likes and subscribers — a portion of which you'd hope would convert into customers. Luckily, this is no longer the only way to use Clarify image for your business. Clarify image has created an API that allows you to integrate your website with the social platform.

Here are some of the greatest benefits of using the Clarify image API:

You can take advantage of existing content. With the Clarify image API, you don't have to spend valuable time and resources creating content. Instead, you can take advantage of the content that has already been published to increase user engagement on your site.

Clarify image is one of the top social media platforms. With more than two billion active users in more than 100 countries, Clarify image is the top image streaming app. It's also the second-

Largest search engine, the second most popular social platform, and the second most visited site on the internet.

Featuring Clarify image's content can increase your dwell time. Because the Clarify image API can help you add Clarify image content to your website, it will help users stay on the page for longer as they watch the content that you've specifically curated for them. This increase in dwell

Time signals to search engines that your content is valuable, boosting your rankings for target keywords.

Using the API can help you convert more website visitors. Using the API is also valuable from a conversion rate and revenue standpoint. Prospects who engage with your content and stay on your website are more likely than not to make a purchase.

Using the Clarify image API is a no-brainer for many marketers, but Clarify image doesn't allow just anyone to access this valuable tool or its data. To prevent potential abuse, Clarify image grants each user a unique API key that allows them to connect their website or app to the platform.

Each API key is designated for an individual project. If you're using the API on two different websites, you should create two different API keys.

5.1 TESTING LEVELS

Software testing is a critical element of software quality assurance and represents the ultimate reuse of specification. Design and code testing represents interesting anomaly for the software during earlier definition and development phase, it was attempted to build software from an abstract concept to tangible implementation.

The testing phase involves, testing of the development of the system using various techniques such as White Box Testing, Control Structure Testing.

6.2 Software Strategies

A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level against customer requirements.

Unit testing: Unit testing tests the functionality of individual units of source code. It is the smallest component of a testable software that works in isolation with other parts of the code. I have done unit testing for various individual components of the source code to uncover errors within the boundary of the application.

Integration testing: Integration testing focuses on the design and construction of the software. Here the individual components that are tested using unit tests are combined and tested as a group. Its primary purpose is to expose the defects associated with the interfacing of modules. It checks if the modules perform the desired functionality when integrated together

System testing: System testing is performed on a completely integrated system to see if it meets the requirements, System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements. In system testing, integration testing passed components are taken as input.

Regression testing: Regression testing aims at verifying the functionality of the software that is previously tested and to which changes are made. It is to ensure the old software still works with new changes.

Acceptance testing: Acceptance testing is conducted to verify if the system compliance the business requirements. Software Testing is a method to check whether the actual

Software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software/system components using manual or

automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements. Adhering to the levels of testing, Unit testing is performed on individual components of the system ensuring the expected behaviour. Later, I have integrated various components Together and performed Integration testing. Once the integration testing is done, I have performed System 30 testing and ensured the application works as per the requirements. Finally, acceptance testing is performed to check if the client accepts the system

- All its module are working properly and the project is best viewed both in Computer.
- The main requirement for this project that you should have good internet connection.

Performance Testing: Performance testing is performed to determine how well the system can perform in terms of responsiveness under all kinds of load. The web application is tested to see if it can sustain huge amount of requests providing higher throughput under different loads. I have simulated multiple hits on various pages of the application to evaluate the overall performance.

5.2 TESTING TECHNIQUES:

5.2.1 White Box Testing:

White box testing is a test case design method that uses the control structure of the procedural design to derive test cases. 88

5.2.2 Control Structure Testing

The following tests were conducted and it was noted that the BCBS is performing them well.

- Basic path Testing
- Condition Testing
- Data Flow Testing
- Loop Testing

There are many type of testing including in white box and black box testing.

5.2.3 Usability Testing:

- Usability testing is nothing but the User-friendliness check.
- In Usability testing, the application flow is tested so that a new user can understand the application easily.

CHAPTER 7

FUTURE SCOPE AND CONCLUSION

Future Scope:

The future of the Clarify image holds immense potential for growth and innovation. Here are some key areas for future development:

1. Enhanced User Experience:

- Implementing advanced algorithms for personalized content delivery based on user preferences and behaviour.
- Integration of augmented reality (AR) and virtual reality (VR) elements to create a more immersive user experience.

2. Advanced Content Moderation:

- Leveraging machine learning and artificial intelligence for more effective and nuanced content moderation, reducing the prevalence of harmful or inappropriate content.
- Incorporating block chain technology to ensure transparency and accountability in content moderation decisions.

3. Monetization Strategies:

- Introduction of innovative monetization features, such as exclusive content subscriptions, premium user accounts, and a decentralized tipping system.
- Exploring partnerships with businesses for targeted advertising and sponsored content.

4. Global Expansion:

- Focusing on expanding user bases in untapped markets and regions to create a truly global social media platform.
- Localization efforts to cater to diverse linguistic and cultural preferences.

5. Interoperability and Integration:

- Facilitating seamless integration with other popular social media platforms and communication tools.
- Exploring interoperability with emerging technologies, such as the integration of decentralized identity systems.

6. Accessibility and Inclusivity:

- Continuous improvements to make the platform more accessible for users with disabilities.
- Implementing features that promote inclusivity and diversity, fostering a safe and welcoming online environment.

7. Environmental Sustainability:

- Adopting eco-friendly practices and technologies to reduce the platform's carbon footprint.
- Encouraging and supporting environmental initiatives within the user community.

8. Open Source Collaboration:

- Opening up certain aspects of the platform for community-driven development through open source initiatives.
- Collaborating with developers to enhance features, security, and overall platform performance.

Conclusion:

In conclusion, the Clarify image stands at the forefront of social media innovation, providing users with a platform for real-time communication and information sharing. As we look to the future, the emphasis will be on continuous improvement, technological advancements, and user-centric innovations. By addressing issues related to content moderation, embracing new monetization strategies, expanding globally, and staying at the forefront of technological trends, the Clarify image is poised to remain a dynamic and influential player in the social media landscape. The commitment to user experience, accessibility, and environmental sustainability ensures that the platform evolves in a responsible and inclusive manner, fostering a global community that thrives on meaningful connections and shared experiences.

BIBLIOGRAPHY

- [1] Introduction To PHP Retrieved on 10/10/2023 from <https://www.tutorialspoint.com/php/index.html>
- [2] [https://www.udemy.com/course/create-a-Clarify image--with-php-mysql-and-js-from-scratch/](https://www.udemy.com/course/create-a-Clarify-image-with-php-mysql-and-js-from-scratch/)
- [3] <https://www.geeksforgeeks.org/php-tutorial/>
- [4] <https://www.javatpoint.com/php-tutorial>
- [5] <https://www.javatpoint.com/javascript-tutorial>