GROUP IRIS

FINAL PROJECT

SCSV 3213 FUNDAMENTAL OF IMAGE PROCESSING

Members : 1. Aimi Binti Rusdi (B19EC0001)
2. Mirhanieza Binti Matharuzaman (A18CS0106)
3. Nuramyra Natasha Binti Ismalludin (B19EC0035)

Lecturer : Dr. MD Sah Hj Salam

« TABLE OF CONTENTS »

</>

01

Project Introduction

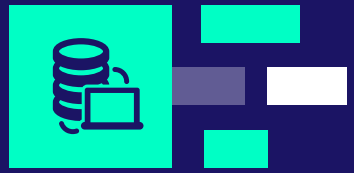
DESCRIPTION »

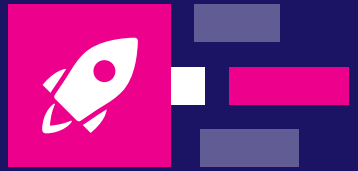
Develop

Simple GUI application

Process

Input image, text or signature with coloured background





Transform

To white background with black or coloured texts

«SITUATION

Useful

- Students who needs their supervisor's signature.
- Stamped remotely
- Via digital documents - white documents.

« METHODS »

01 Masking

02 ROI Operation

03 Image Filtering

04 Heuristics Approach

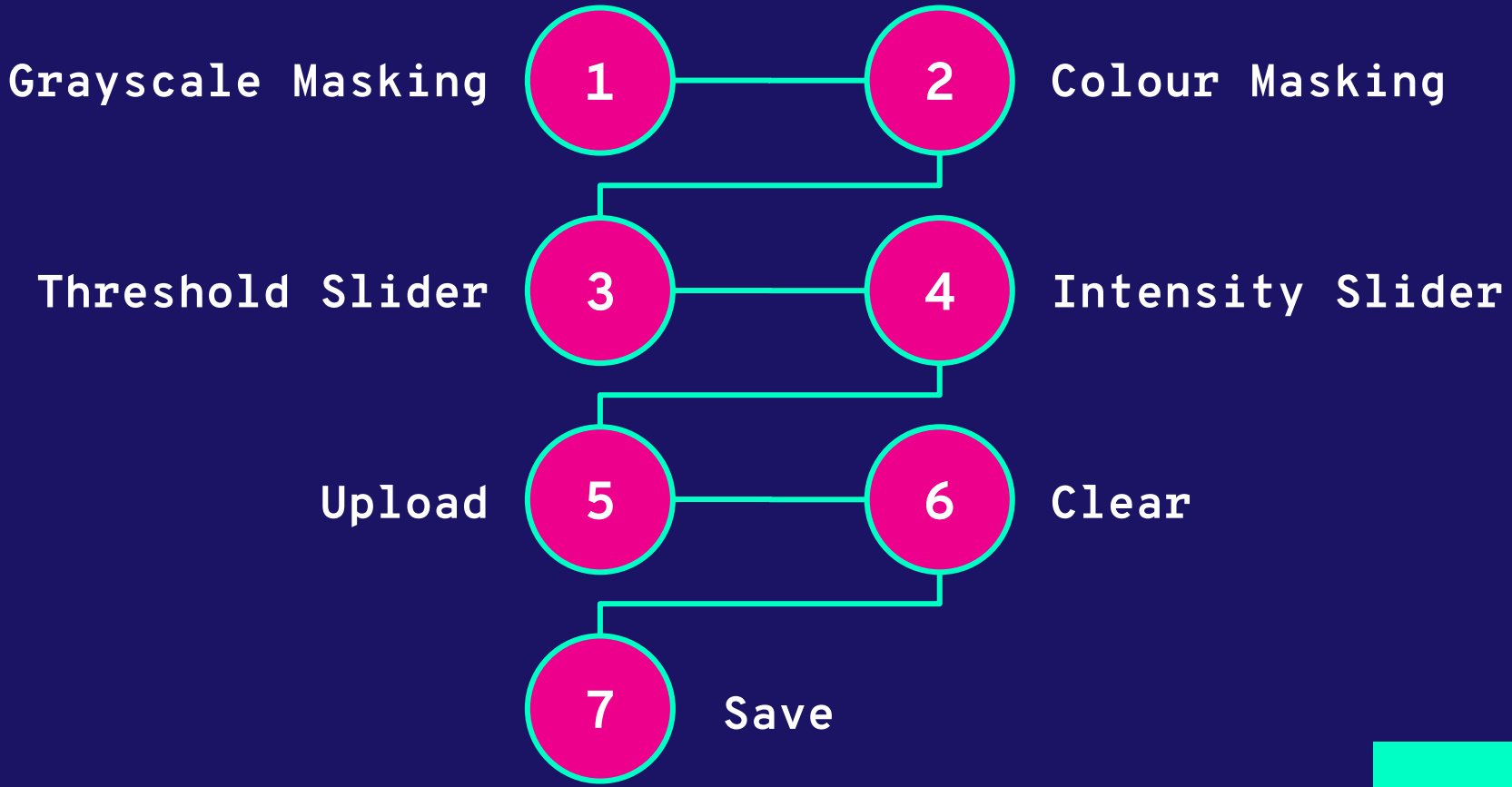
05 Global Image Thresholding

</>

02

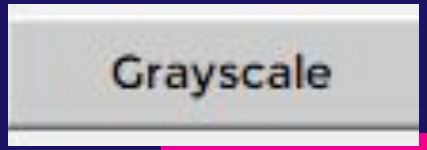
Coding Description

CODING EXPLANATION



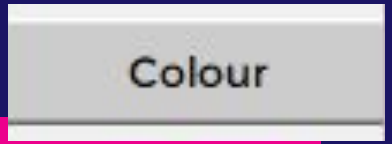
GRAYSCALE MASKING

```
% -------- Executes on button press in grayscale -----------
function grayscale_Callback(hObject, eventdata, handles)
% hObject    handle to grayscale (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)
global T;
global img;
global nimg;
try
    %obtain grayscale img
    imgray = rgb2gray(img);
    %segmented img
    I_tresh = im2bw(img,(T/255));
    %convert segmented img into uint8
    I_tresh=im2uint8(I_tresh);
    %combine segmented and grayscale img
    not_roi = bitor(imgray,I_tresh);
    nimg = not_roi;
    imshow(not_roi,'Parent',handles.axes2);
catch me
end
```



- Original image is first converted into grayscale
- Uses **Global Thresholding** for value *T*.
- Image thresholding uses **im2bw** to transform the image to black and white image.
- Uses **bitor** operation to mask combination of grayscale and segmentation images as the mask.

COLOUR MASKING



- Uses **Global Thresholding** for value *T*.
- Image thresholding uses **im2bw** to transform the image to grayscale.
- From detection of selected region, uses **immultiply 1.5** to enhance selected colours.

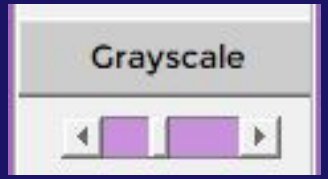
```
% ------------ Executes on button press in colour ----------
function colour_Callback(hObject, eventdata, handles)
% hObject    handle to colour (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)
try
global img;
global T;
global nimg;
    I = img;
    I_tresh=im2bw(I,(T/255));
    I_tresh=im2uint8(I_tresh);
    not_roi = bitor(I,I_tresh);
    not_roi = immultiply(not_roi,1.5);
    imshow(not_roi,'Parent',handles.axes2);
    nimg = not_roi;
    % Update handles structure
    guidata(hObject, handles);
catch me
end
```

THRESHOLD SLIDERS

```
% ---------- Executes on slider movement for grayscale --------------
function grayscale_slider_Callback(hObject, eventdata, handles)
% hObject    handle to grayscale_slider (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range
of slider
global img;
global T;
global nimg;
    greyimg = rgb2gray(img); %obtain grayscale img
    assignin('base','T',T);
    T = get(hObject,'Value'); %threshold value becomes slider value
    %set current position of slider
    set(handles.grayscale_slider, 'Value', T);
    i_tresh = im2bw(img,(T/255)); %obtain segmented img
    i_tresh = im2uint8(i_tresh); %convert into uint8
    %combine segmented and grayscale img
    not_roi = bitor(greyimg,i_tresh);
    imshow(not_roi ,'Parent',handles.axes2);
    nimg = not_roi;
    % Update handles structure
    guidata(hObject, handles);

    % Update handles structure
    guidata(hObject, handles);
```



- Threshold value **T** is set with slider value
- The conditions must be set to the specific slider handles.
- Uses **bitor** operation to mask image and adjust colour.

THRESHOLD SLIDERS

- Threshold value **T** is set with slider value
- Assign the threshold value into the slider value for adjustments.
- Set at the correct slider similar to grayscale.

```
% ------------- Executes on slider movement for colour --------------
function colour_slider_Callback(hObject, eventdata, handles)
% hObject    handle to colour_slider (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range
of slider
try
    global img;
    global nimg;
    global T;
    I = img;
    T = get(hObject,'Value');
    assignin('base','T',T);
    set(handles.colour_slider, 'Value', T);
    i_tresh = im2bw(I,(T/255));
    i_tresh = im2uint8(i_tresh);
    not_roi = bitor(I, i_tresh);
    not_roi = immultiply(not_roi,1.5);
    nimg = not_roi;
    imshow(not_roi,'Parent',handles.axes2);

    % Update handles structure
    guidata(hObject, handles);
catch me
end
```

INTENSITY SLIDERS

```
% ----------- Executes on slider movement for intensity -------------
function intense_slider_Callback(hObject, eventdata, handles)
% hObject    handle to intense_slider (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range
of slider
global nimg;
    nimg = im2uint8(nimg);
    offset = get(hObject,'Value');
    nimg2 = imadjust(nimg,[offset  1],[]);
    assignin('base','offset',offset);
    set(handles.intense_slider, 'Value', offset);
    %nimg = imadd(nimg,offset);
    imshow(nimg2,'Parent',handles.axes2);
    % Update handles structure
    guidata(hObject, handles);
```

- **Im2uint8** converts an image into uint8 class which is 8bit.
- The conditions must be set to the specific slider handles.
- **Low-in** value of the imadjust() is manipulated.

UPLOAD

- **imread** is to read the selected image file.
- Set as the current data for the application using handles.
- Display at both axes, one for original and other for edit.

```
% ------------ Executes on button press in upload ----------
function upload_Callback(hObject, eventdata, handles)
% hObject    handle to upload (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
global T;
try
    [a b] = uigetfile('*.*','All Files');
    img = imread([b a]);
    handles.currentData = img;
    imshow(handles.currentData,'Parent',handles.axes1);
    imshow(handles.currentData,'Parent',handles.axes2);
    setImg(img);

    T = 100;
    % Update handles structure
    guidata(hObject, handles);

    catch e
        f = warndlg("Something happened while uploading!",
"Error");
end
```

CLEAR

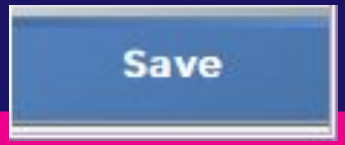
</>

```
% ----------- Executes on button press in clear ------------
function clear_Callback(hObject, eventdata, handles)
% hObject    handle to clear (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)
try
    cla(handles.axes1);
    cla(handles.axes2);
    set(handles.axes1,'visible', 'off')
    set(handles.axes2,'visible', 'off')
catch me
end
```

</>

- Selection of handles onto which should be disabled or not.
- Axes 1 and axes 2 box's visibility is turned off.
- The selected image will disappear.

SAVE



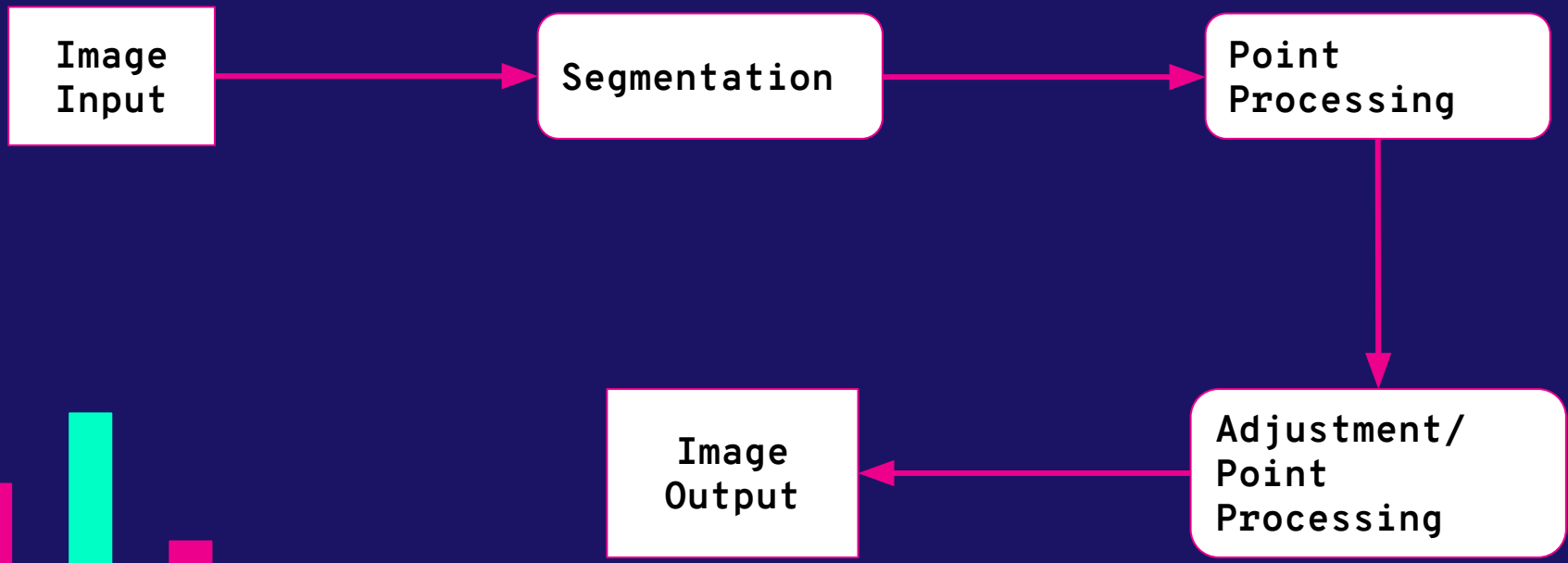
- **imwrite** is to create new file in your folder.
- It is set as for Transparency image with background masking.
- **[1 1 1]** will set white colour as transparent.

```
% -------------- Executes on button press in save -------------
function save_Callback(hObject, eventdata, handles)
% hObject    handle to save (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
try
    frame = getframe(handles.axes2);
    img = frame2im(frame);
    [filename, foldername] = uiputfile('*.png', 'Where do you
want the file saved?');
    fname = fullfile(foldername, filename);
    imwrite(img, fname, 'Transparency', [1 1 1]);
    % Update handles structure
    guidata(hObject, handles);
    catch e
        f = warndlg("Something happened while
saving!","Error");
end
```

</>

03

Pipeline Process

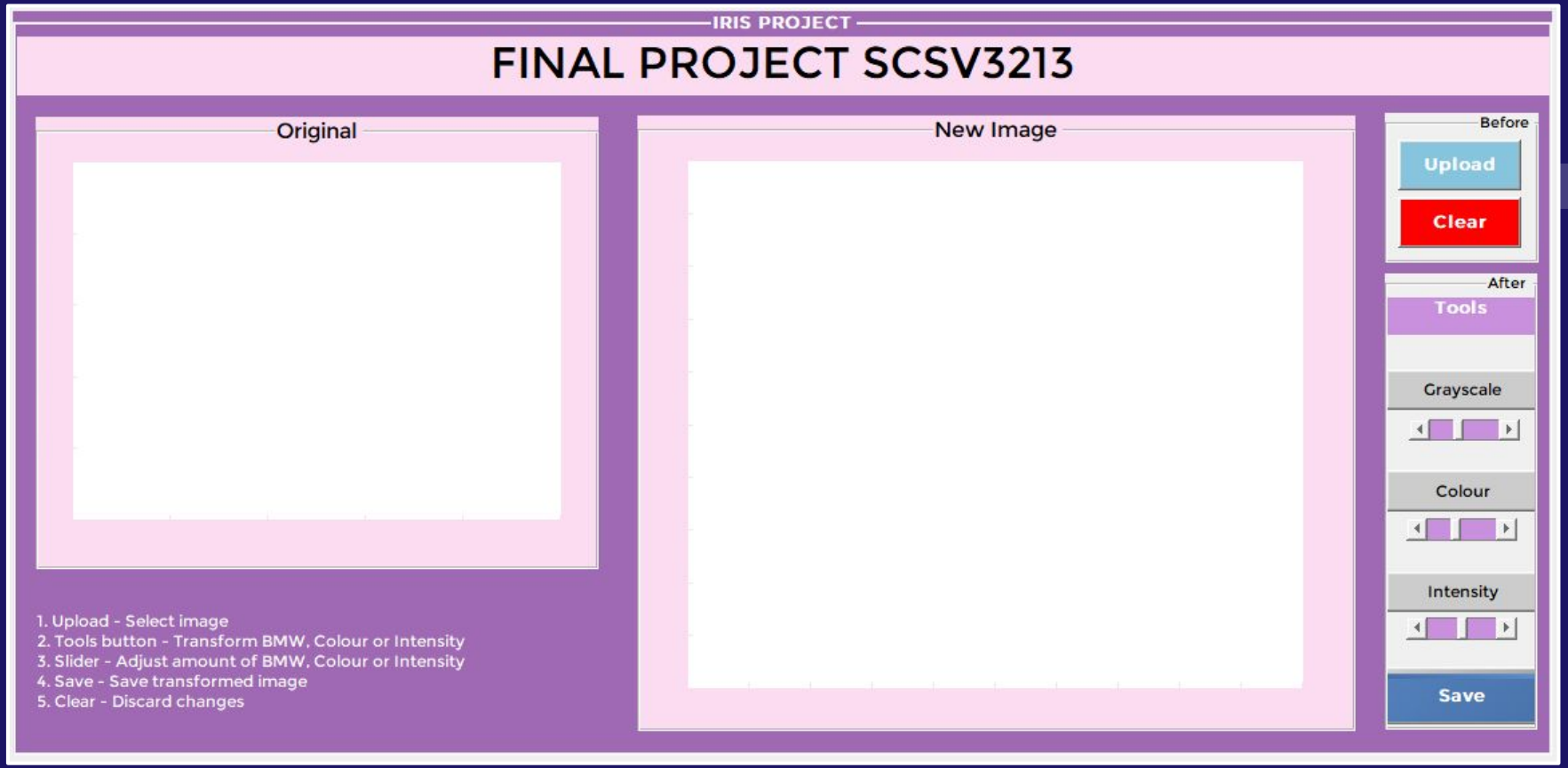

Heuristic Threshold
Logic Operation OR
Image Input
Segmentation
Point Processing
Adjustment/ Point Processing
Image Output
Multiplication
Histogram Processing (imadjust)

</>

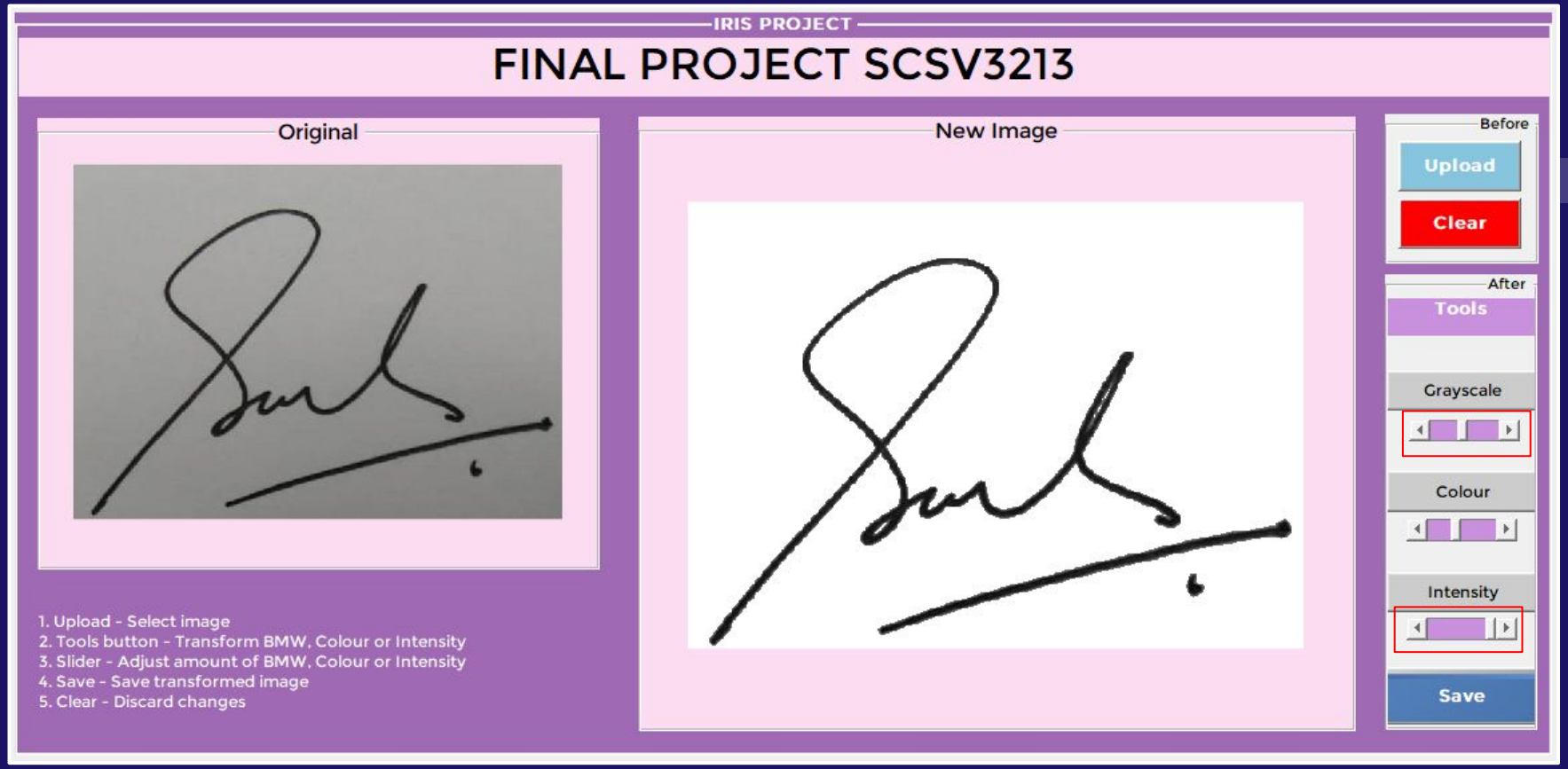
04

Input and Output

IRIS GUI APPS



GRAYSCALE IMAGE



COLOURED IMAGE

COLOURED INTENSITY IMAGE

IRIS PROJECT

FINAL PROJECT SCSV3213

Original

New Image

Before

Upload

Clear

After

Tools

Grayscale

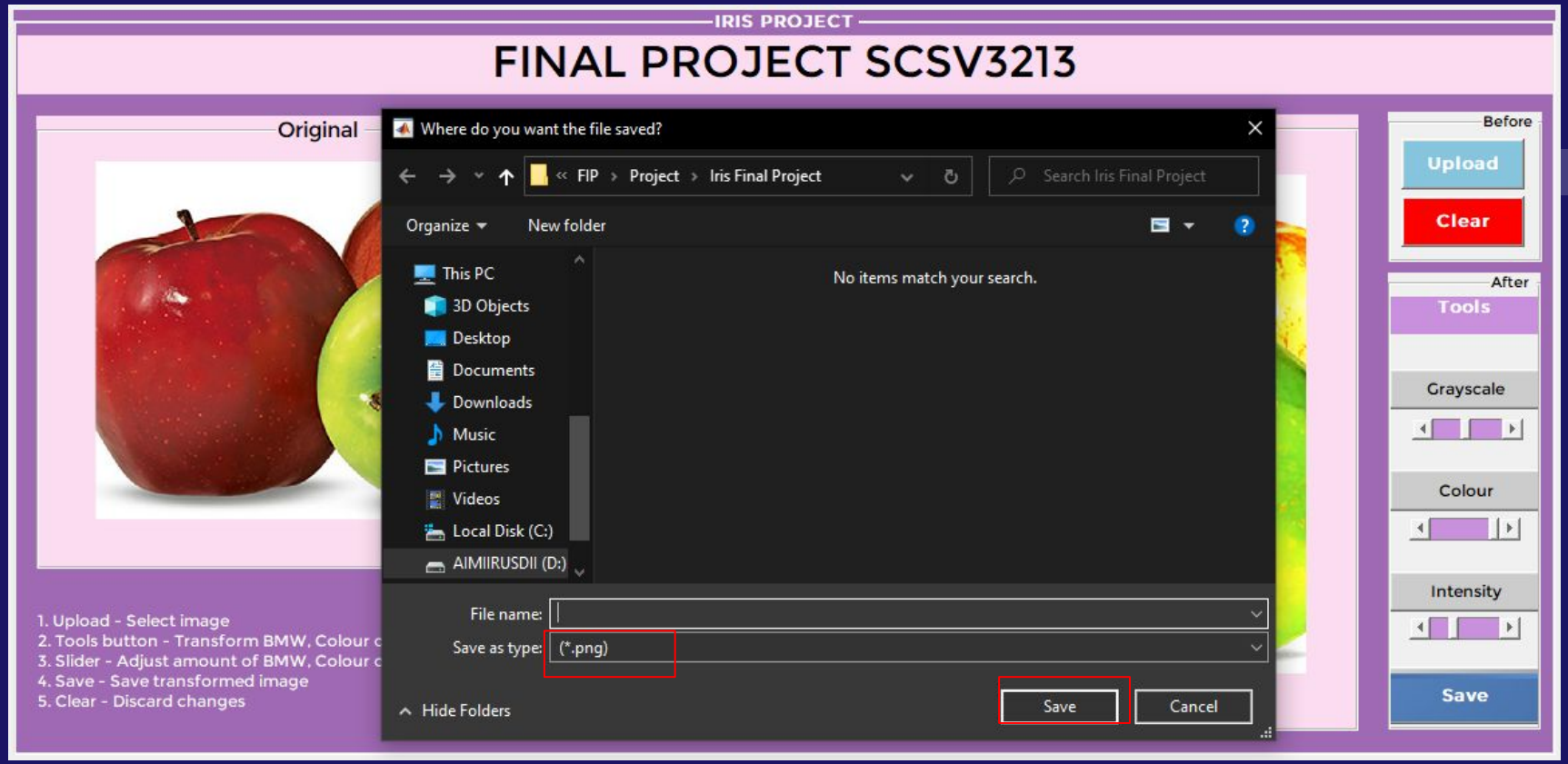
Colour

Intensity

Save

1. Upload - Select image
2. Tools button - Transform BMW, Colour or Intensity
3. Slider - Adjust amount of BMW, Colour or Intensity
4. Save - Save transformed image
5. Clear - Discard changes

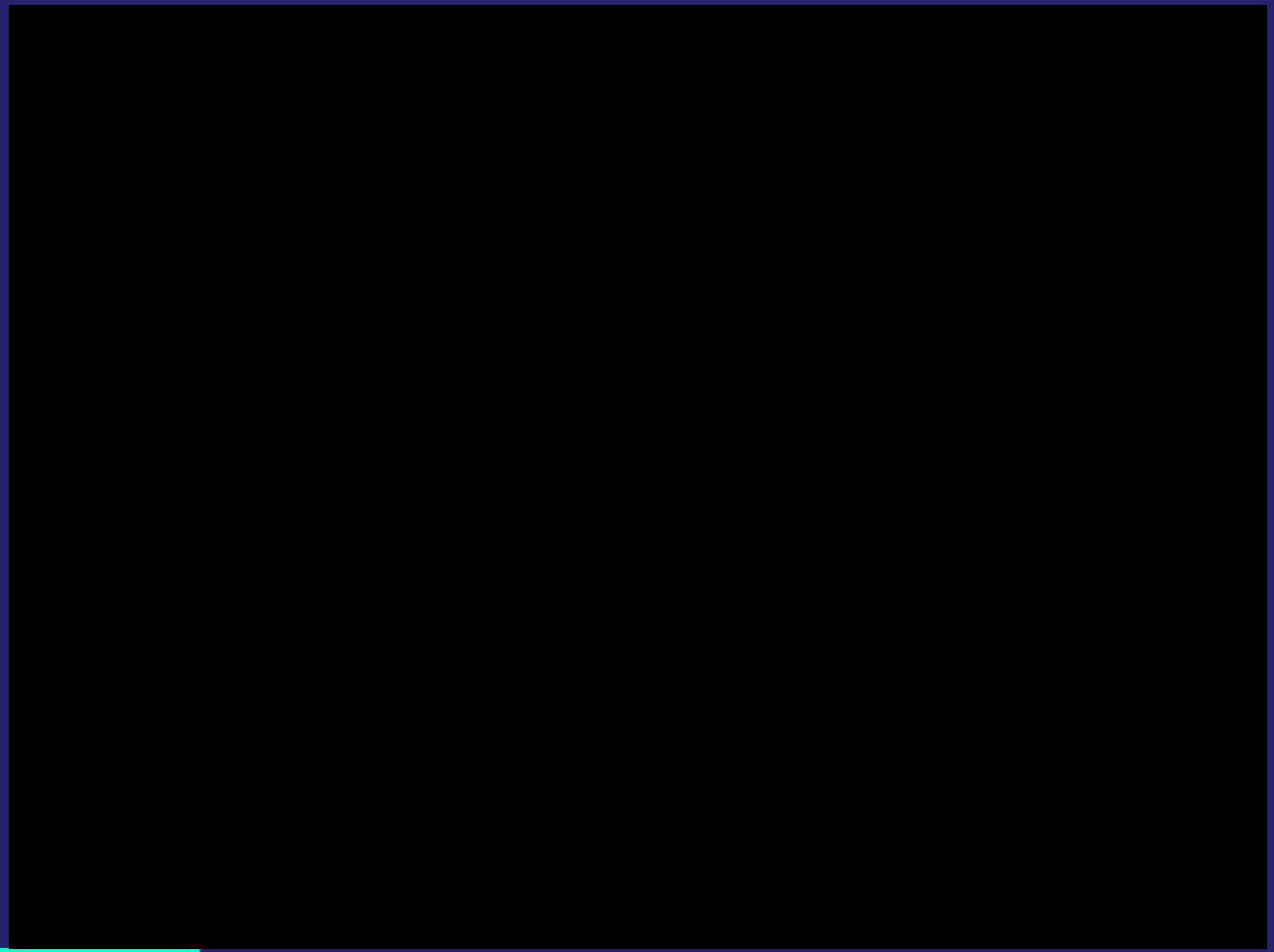
SAVE IMAGE


IRIS PROJECT
FINAL PROJECT SCSV3213
Original
Where do you want the file saved?
FIP › Project › Iris Final Project
Search Iris Final Project
Organize
New folder
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
AIMIIRUSDII (D:)
No items match your search.
File name:
Save as type: (*.png)
Hide Folders
Save
Cancel
Before
Upload
Clear
After
Tools
Grayscale
Colour
Intensity
Save
1. Upload - Select image
2. Tools button - Transform BMW, Colour c
3. Slider - Adjust amount of BMW, Colour c
4. Save - Save transformed image
5. Clear - Discard changes

</>

05

User GUI DEMO



</>

06

Conclusion

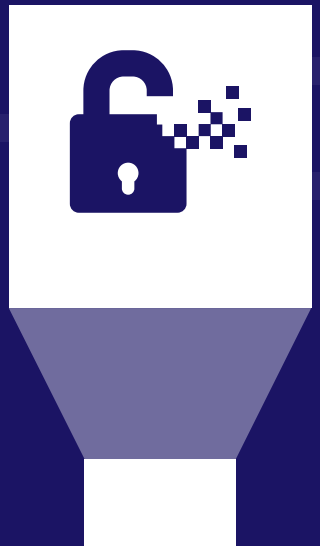
FEATURES OF THE PROJECT

Application

Remove the background image to transparent or white

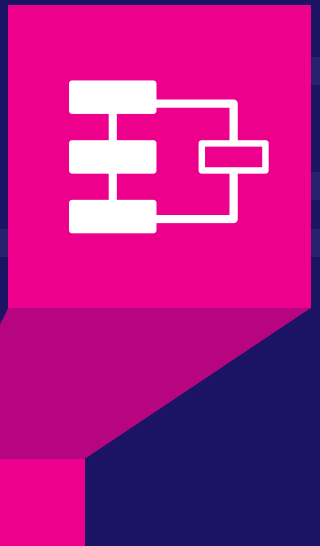
Apply

Image masking is applied to modify image according to colours.

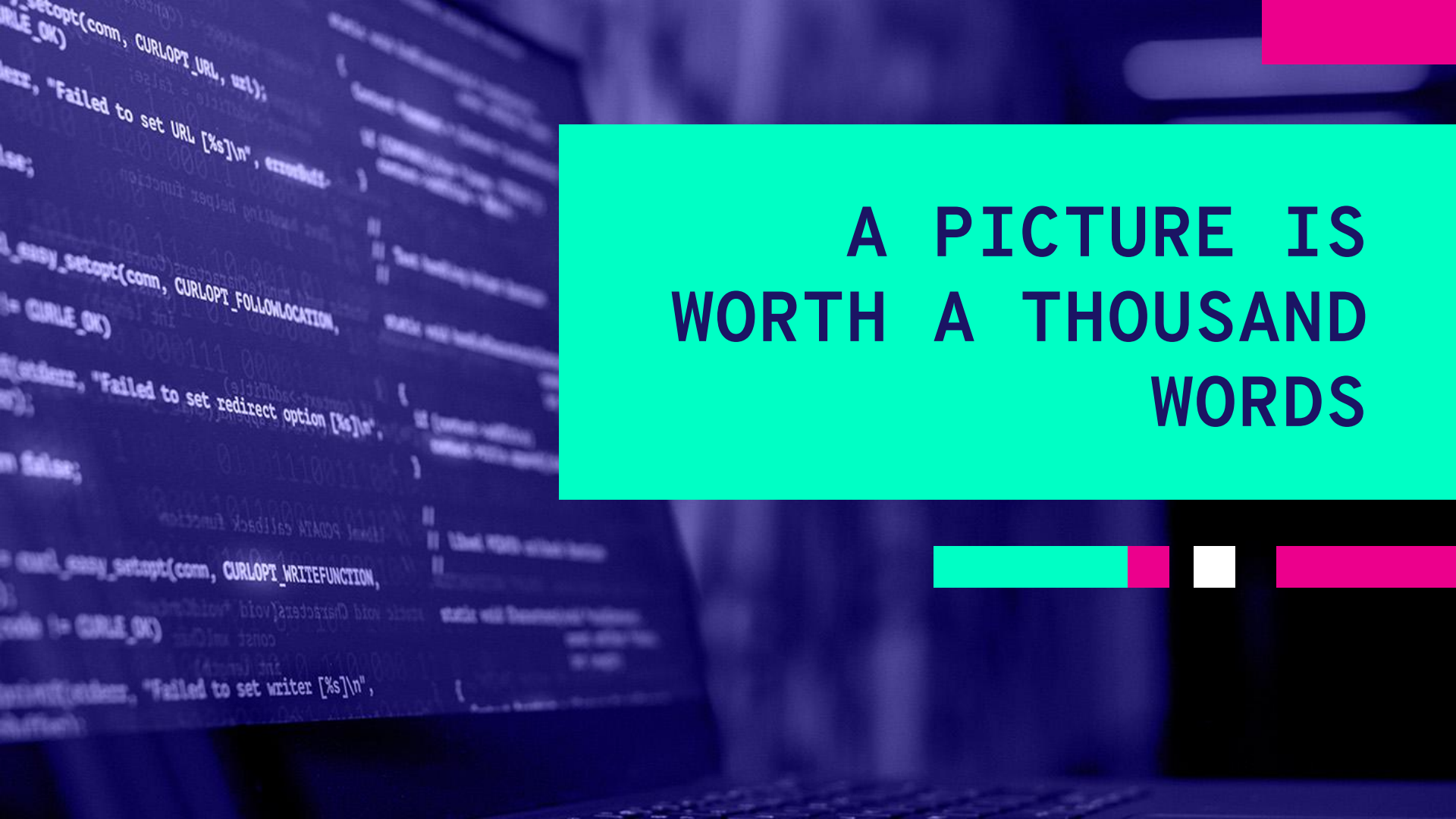


Partition

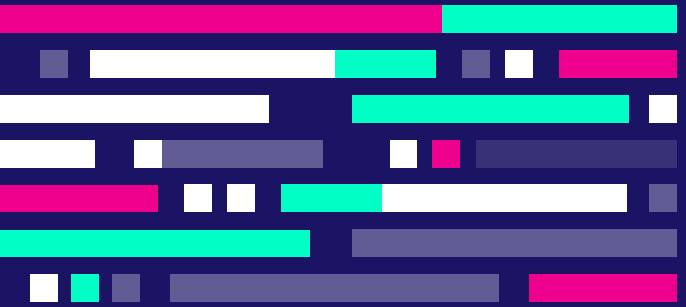
Image segmentation selects or segments areas based on pixels.




THANK YOU.


A PICTURE IS
WORTH A THOUSAND
WORDS

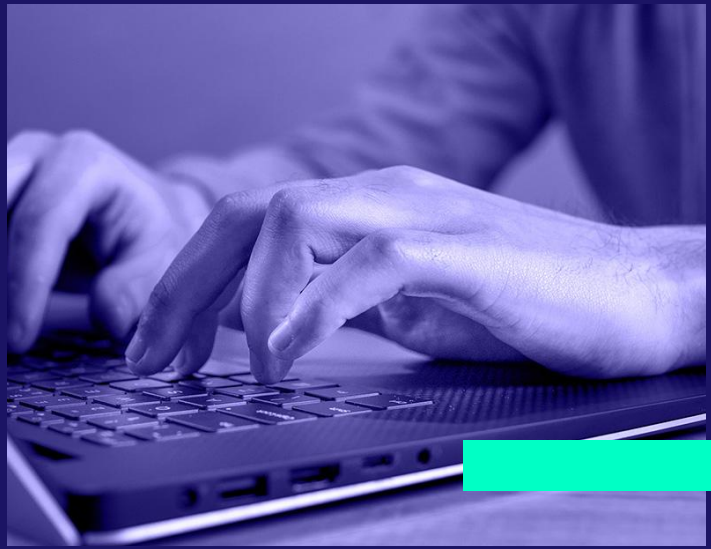
Project Introduction



- Develop a simple application.
- Process input image, text or signature
  - Coloured or not a white background
- Transform to a white background
  - Image becomes black or coloured.
- Situation given :-
  - Student needs their supervisor's signature stamped remotely.
- **Methods used :-**
  - Segmentation
  - Masking
  - Heuristics
  - ROI Operation
  - Addition

INTRODUCTION

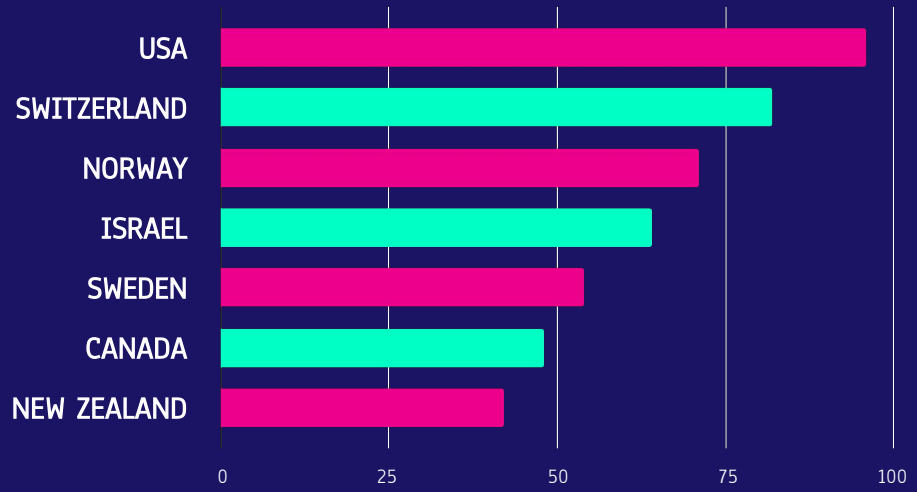
Mercury is the closest planet to the Sun and the smallest one in the Solar System—it's only a bit larger than the Moon. The planet's name has nothing to do with the liquid metal



“This is a quote, words full of wisdom that someone important said and can make the reader get inspired.”

—SOMEONE FAMOUS

DID YOU KNOW THIS?



Average Software Developer Salaries in the World

$96,290

programming salary range

-4.30 %

ASP.NET usage is decreasing

5,500,000

active PHP developers

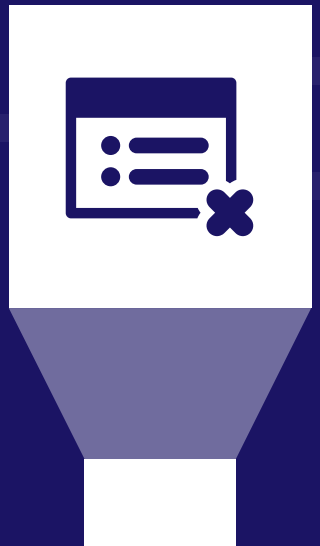
FEATURES OF THE TOPIC

Mercury

Mercury is the closest planet to the Sun

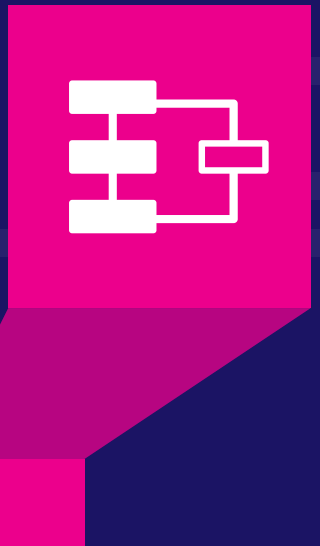
Venus

Venus is the second planet from the Sun



Saturn

Saturn is a gas giant and has several rings



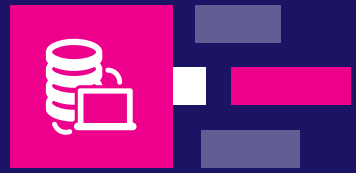
DEFINITION OF CONCEPTS

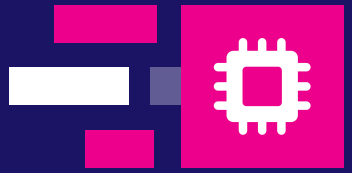
Mercury

Mercury is the closest planet to the Sun

Venus

Venus is the second planet from the Sun



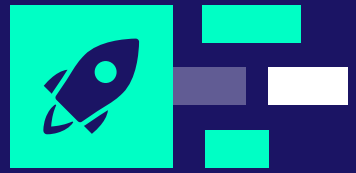


Neptune

Neptune is the farthest planet from the Sun

Saturn

Saturn is a gas giant and has several rings



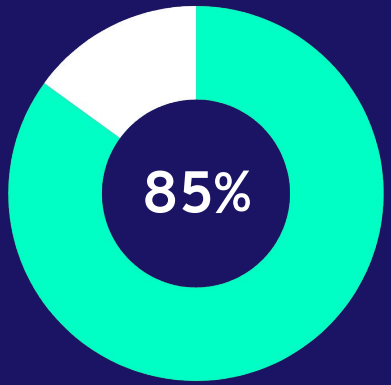
20.4

million active developers use JavaScript over other programming languages

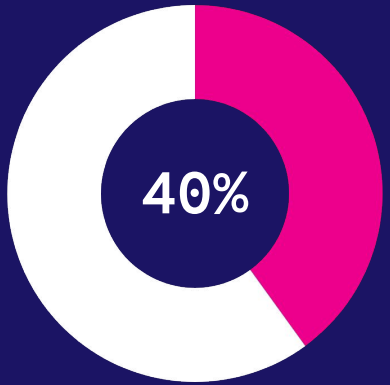
THIS IS A TABLE

	Mass	Diameter	Gravity
Android	1.25	0.50	13.2
HTML5	3.20	2.75	20.5
Python	50.5	10.5	12.4

USES OF THE SUBJECT

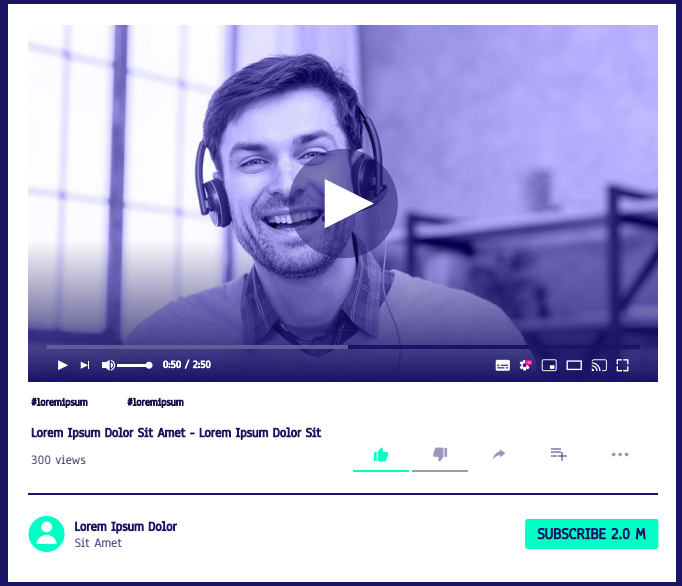


Venus has a beautiful name



Mercury is the smallest planet

DEMO



You can replace the image on the screen with your own work. Just move the filter aside, delete this picture, add yours and place the filter on top again

THIS IS A MAP

Mars

Despite being red,
Mars is a cold place

Saturn

Saturn is a gas giant
and has several rings

PROCESS

Step 1

Mercury is the closest planet to the Sun

Step 2

Venus is the second planet from the Sun

Step 3

Despite being red, Mars is a cold place

Step 4

It's the biggest planet in the Solar System

Step 5

Saturn is the ringed one and a gas giant

Step 6

It's the farthest planet from the Sun

Problem

Jupiter is the biggest planet in the Solar System

Solution

Despite being red, Mars is actually a cold place

OVERVIEW DIAGRAM

Header
Mercury is the smallest planet

Nav
Despite being red, Mars is cold

Body
Earth is where we live on

Footer
Pluto is now a dwarf planet

01

Project Introduction

« TABLE OF CONTENTS »

OTHER CONCEPTS

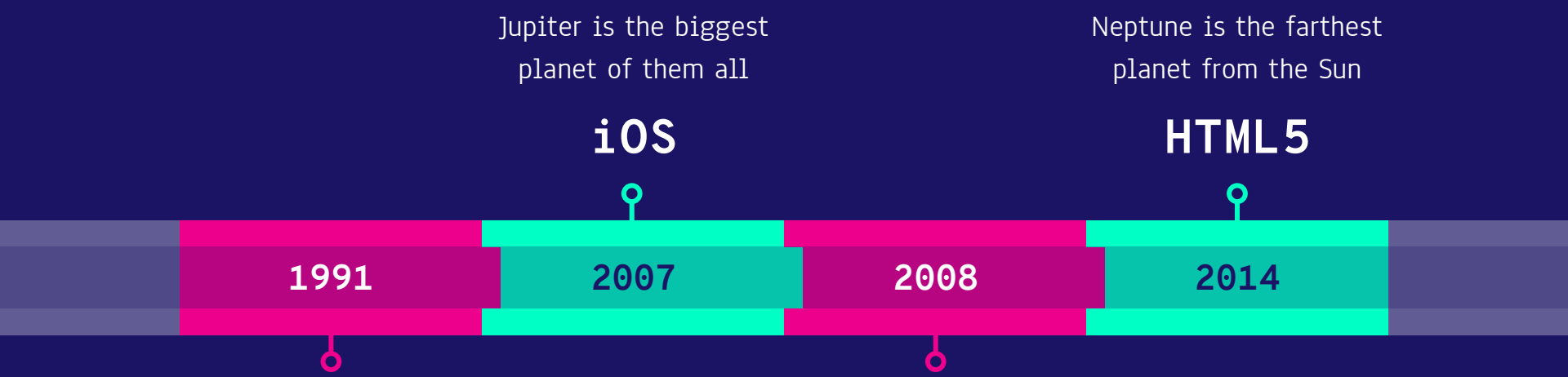
Mercury

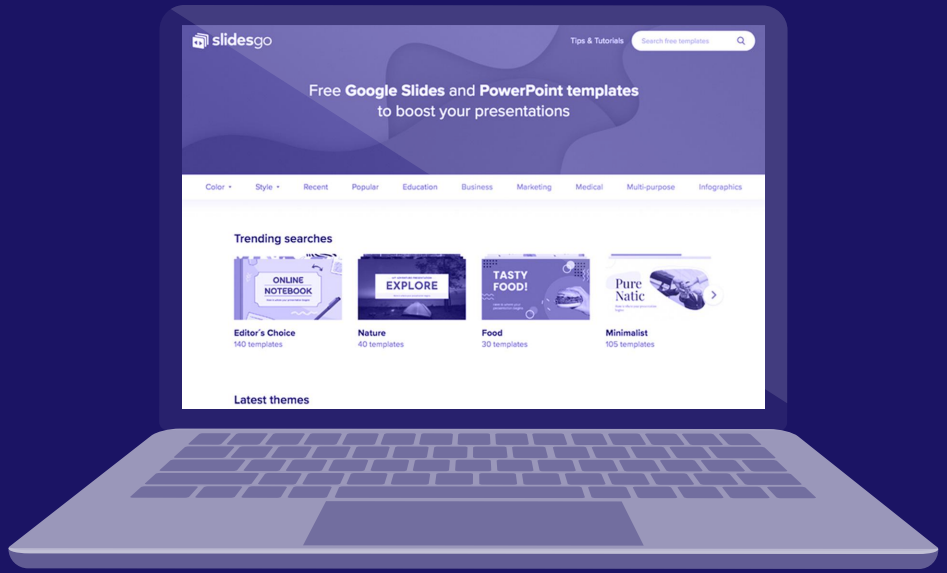
It's the closest planet to the Sun and the smallest in the Solar System

Venus

Venus has a beautiful name and is the second planet from the Sun

PROGRAMMING TIMELINE





SOCIAL MEDIA

You can replace the image on the screen with your own work. Just delete this one, add yours and center it properly

EXERCISE

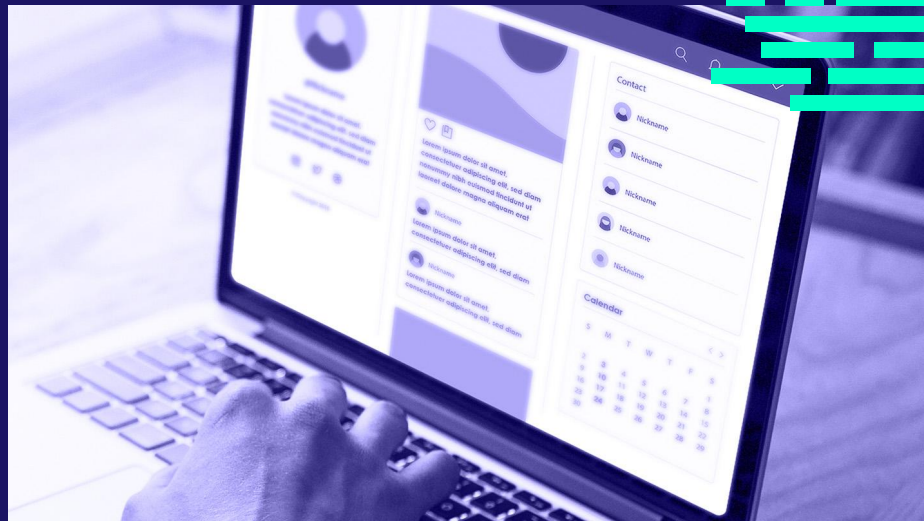
```
<!DOCTYPE html>
<html>
<body>

<canvas id="myCanvas"
width="250" height="25"
style="border:4px solid #00ffc5;">
</canvas>

</body>
</html>
```

ASSIGNMENT

What language is used to render 2D graphics in HTML5?



THANKS !

Do you have any questions?
youremail@freepik.com
+91 620 421 838
yourcompany.com

CONTENTS OF THIS TEMPLATE

Here's what you'll find in this **Slidesgo** template:

1. A slide structure based on a lesson, which you can easily adapt to your needs. For more info on how to edit the template, please visit **Slidesgo School** or read our **FAQs**.
2. An assortment of pictures that are suitable for use in the presentation can be found in the **alternative resources** slide.
3. A **thanks** slide, which you must keep so that proper credits for our design are given.
4. A **resources** slide, where you'll find links to all the elements used in the template.
5. **Instructions for use**.
6. Final slides with:
    - The **fonts and colors** used in the template.
    - A selection of **illustrations**. You can also customize and animate them as you wish with the online editor. Visit **Stories by Freepik** to find more.
    - More **infographic resources**, whose size and color can be edited.
    - Sets of **customizable icons** of the following themes: general, business, avatar, creative process, education, help & support, medical, nature, performing arts, SEO & marketing, and teamwork.

You can delete this slide when you're done editing the presentation.

ALTERNATIVE RESOURCES

PHOTOS:

- Close up of hacker
- Teacher talking with his students online
- Hacking concept
- Young woman enjoying new technologies
- Close up of hacker
- Man using laptop in cafe

RESOURCES

Did you like the resources on this template?
Get them for free at our other websites.

VECTORS:

- Landing page template with programming concept
- Development Icon Pack
- Realistic multimedia player
- Concept of flat computer engineering

Instructions for use

In order to use this template, you must credit **Slidesgo** by keeping the **Thanks** slide.

You are allowed to:

- Modify this template.
- Use it for both personal and commercial projects.

You are not allowed to:

- Sublicense, sell or rent any of Slidesgo Content (or a modified version of Slidesgo Content).
- Distribute Slidesgo Content unless it has been expressly authorized by Slidesgo.
- Include Slidesgo Content in an online or offline database or file.
- Offer Slidesgo templates (or modified versions of Slidesgo templates) for download.
- Acquire the copyright of Slidesgo Content.

For more information about editing slides, please read our FAQs or visit Slidesgo School:

https://slidesgo.com/faqs and https://slidesgo.com/slidesgo-school

Fonts & colors used

This presentation has been made using the following fonts:

Overpass Mono
(https://fonts.google.com/specimen/Overpass+Mono)

Anaheim
(https://fonts.google.com/specimen/Anaheim)

#1b1464

#ffffff

#00ffc5

#ec008c

Stories by Freepik

Create your Story with our illustrated concepts. Choose the style you like the most, edit its colors, pick the background and layers you want to show and bring them to life with the animator panel! It will boost your presentation. Check out How it Works.

Pana

Amico

Bro

Rafiki

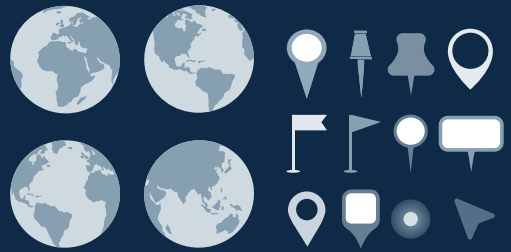
Cuate

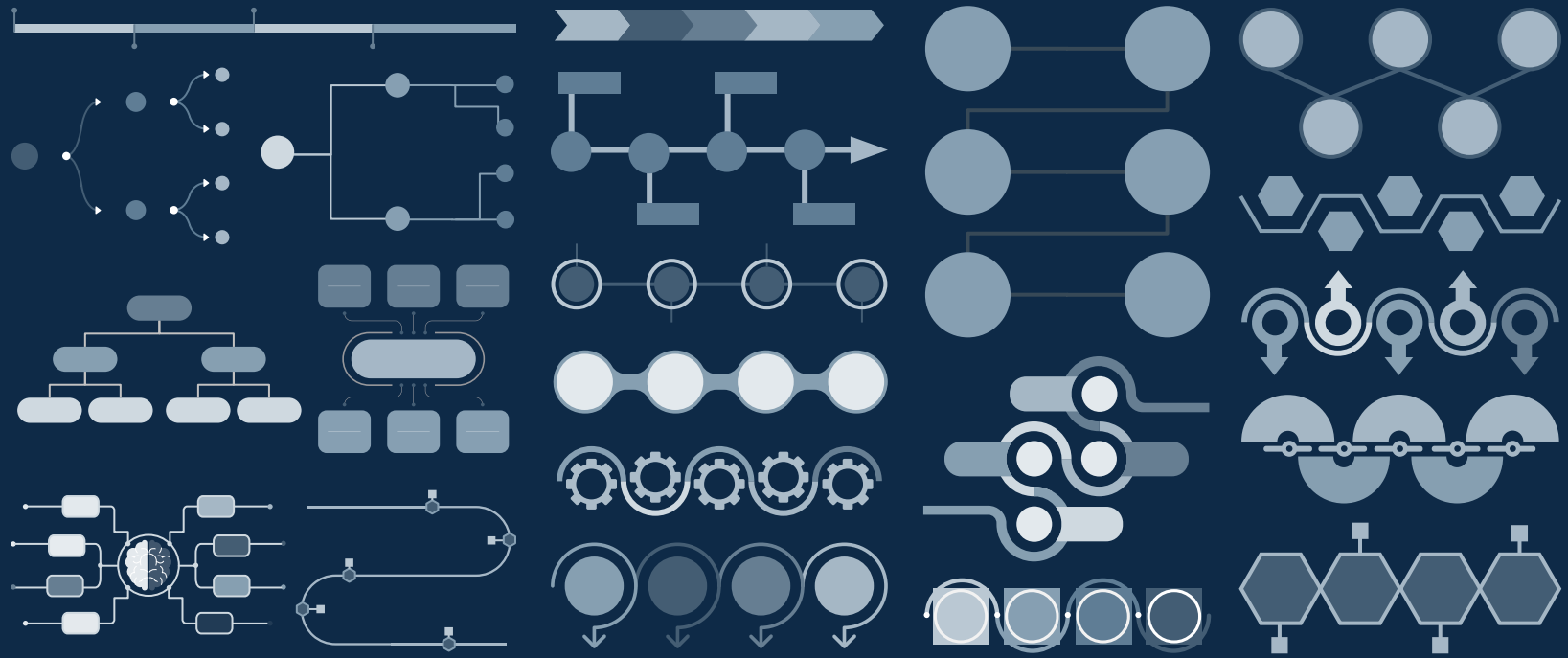
Use our editable graphic resources...

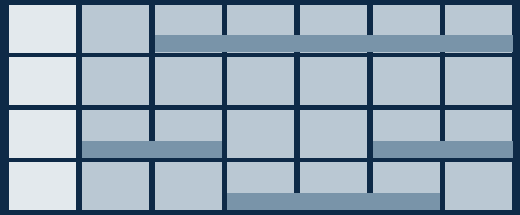
You can easily resize these resources without losing quality. To change the color, just ungroup the resource and click on the object you want to change. Then, click on the paint bucket and select the color you want. Group the resource again when you're done. You can also look for more infographics on Slidesgo.

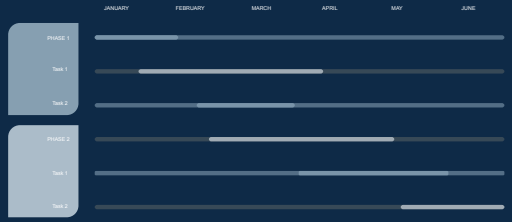


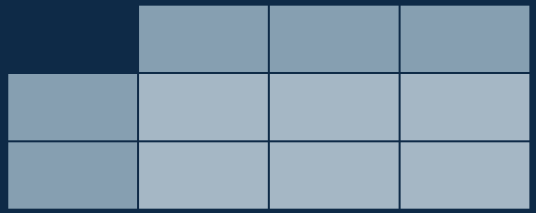


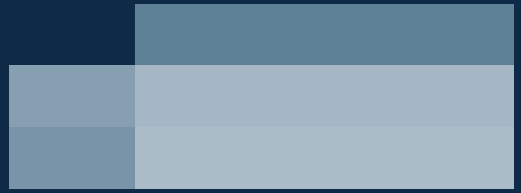



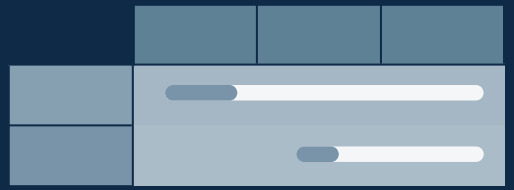


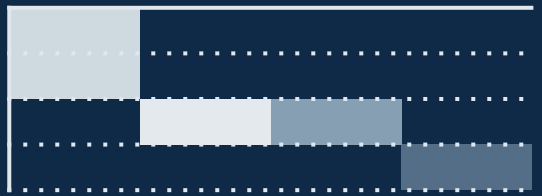


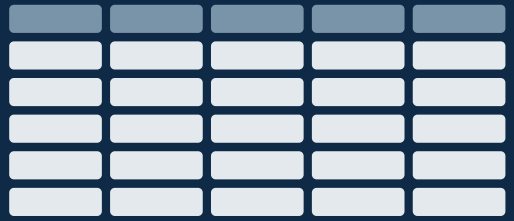

JANUARY
FEBRUARY
MARCH
APRIL
MAY
JUNE
PHASE 1
Task 1
Task 2
PHASE 2
Task 1
Task 2

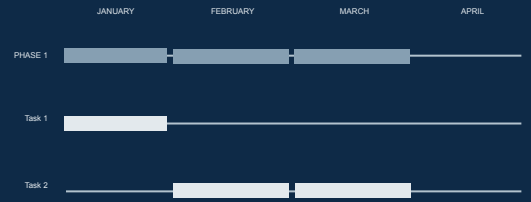


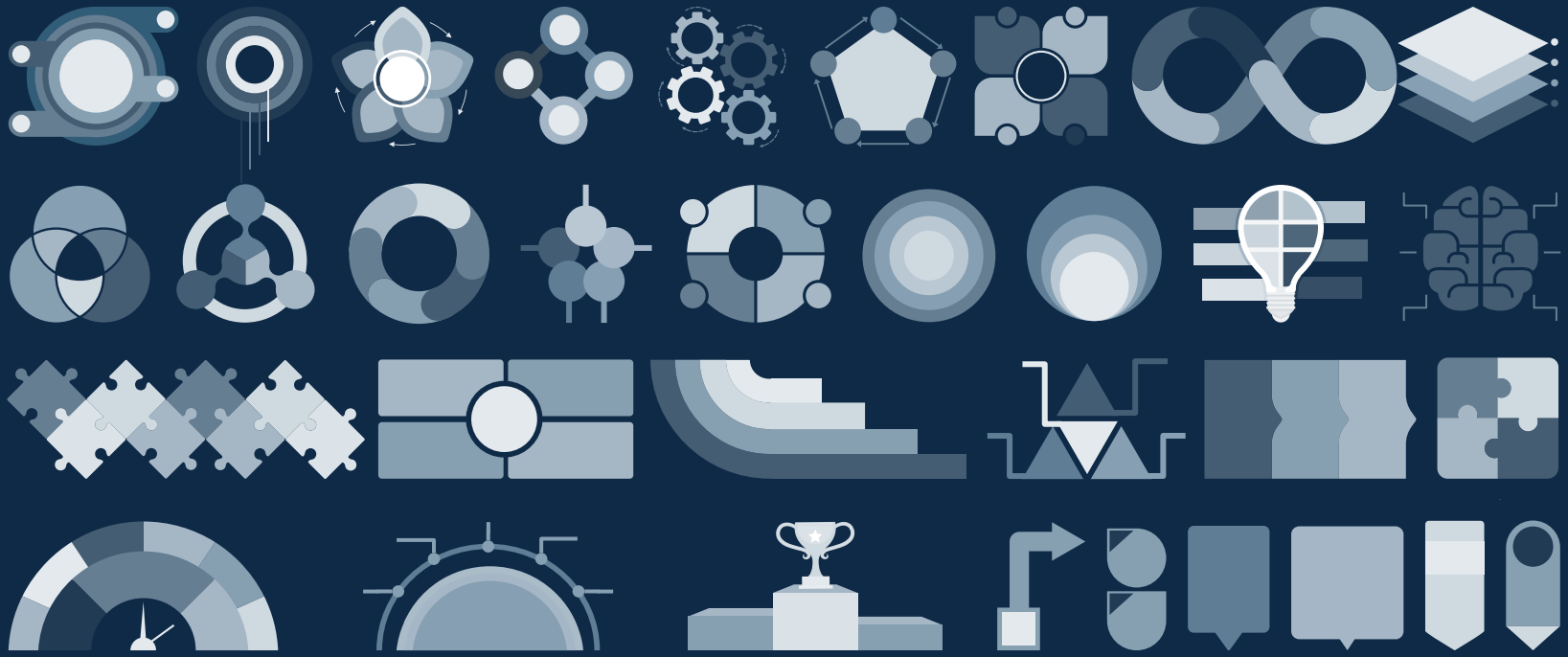


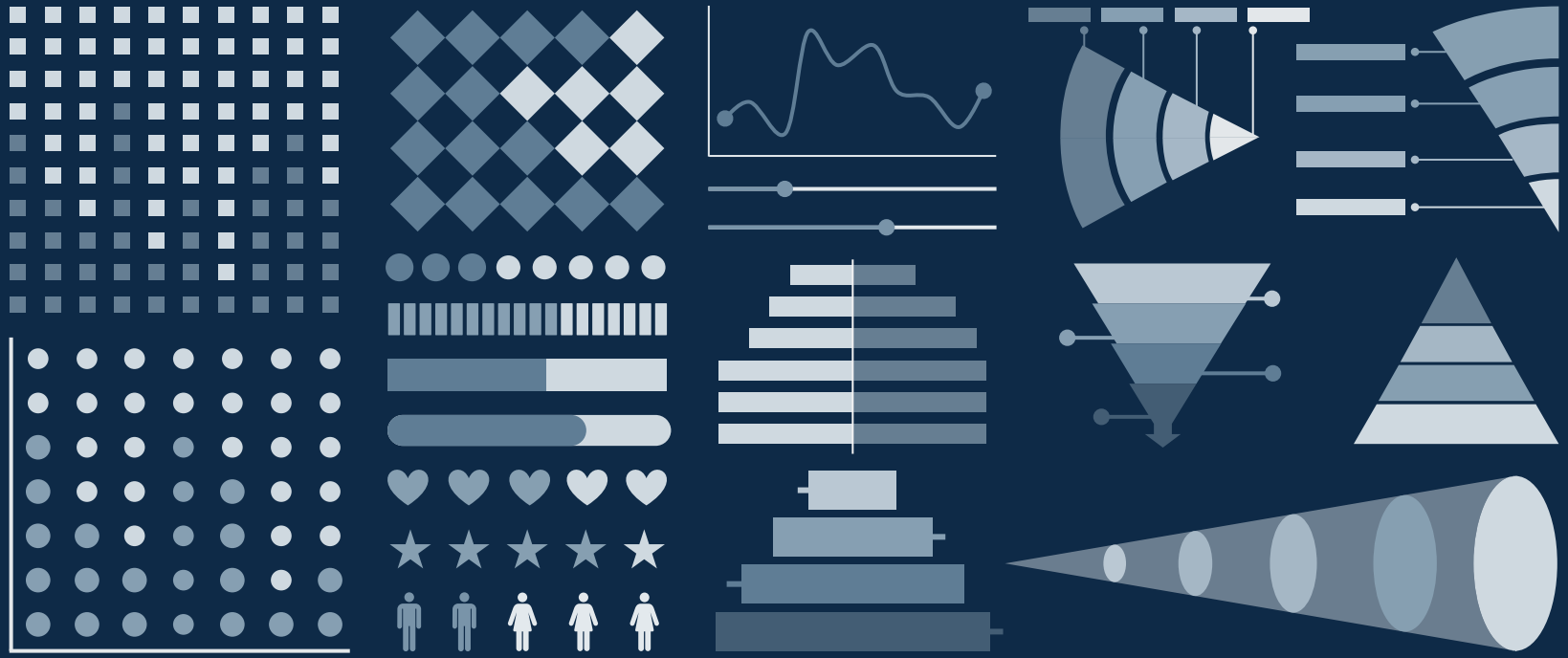







JANUARY
FEBRUARY
MARCH
APRIL
PHASE 1
Task 1
Task 2





...and our sets of editable icons

You can resize these icons without losing quality.
You can change the stroke and fill color; just select the icon and click on the paint bucket/pen.
In Google Slides, you can also use Flaticon's extension, allowing you to customize and add even more icons.

Educational Icons

Medical Icons

Business Icons

Teamwork Icons

Help & Support Icons

Avatar Icons

Creative Process Icons

Performing Arts Icons

Nature Icons

SEO & Marketing Icons


slidesgo