# CSE611 - Photo Editor Tool (Sponsored by University at Buffalo)

*Guided by Professor - JinJun Xiong, TA - Amir Nassereldine*

*Team Members*

*Mallikharjuna Rao Annam, Rakesh Kumar Gavara, Manikanta Kalyan Gokavarapu,Khyathi Kunuthur, Ruthika Juluri*

# 1. Introduction

## 1.1 Overview

The Photo Editor tool is a web application designed to enhance and manipulate digital images. It allows users to edit and enhance their photos with a variety of tools and features, including cropping, resizing, adjusting brightness, contrast, collage, mosaic, merge images into a pdf, background change and noise removal. The software is user-friendly, offering an intuitive interface and the ability to undo or redo actions. The project is aimed at individuals who want to improve their photos or use them for personal or professional purposes. The end goal of the project is to provide users with a comprehensive tool that can help them create high-quality images with ease.

The photo editor project was created with the goal of providing individuals with a convenient and accessible tool for enhancing and manipulating their digital photos. This project aims to bridge the gap between basic photo editing tools and more advanced image editing software, offering a comprehensive set of features in a user-friendly interface. Target audience includes individuals, hobbyists, amateur photographers, professionals, students and businesses(small and big).During development, the team faced constraints such as balancing the need for advanced features with user-friendliness. Other constraints the team faced include the type of algorithms for face detection, balance between the photo quality and time taken for image processing.

## 1.2 Problem Statement

The problem statement is that while there are many photo editing tools available in the market, including Canva, Ribbet, BeFunky, Adobe Photoshop Express, and PicsArt, None of these applications provide a complete range of features, including the creation of passport photos with face, pose, spectacles detection, and background change to comply with various country requirements. Additionally, none of these applications offer Mosaic and image format conversion services. Some applications do offer these services individually, but they are mostly paid and filled with advertisements, limiting the user's ability to freely use the software and also there exist numerous security issues associated with the images we upload. Therefore, there is a need for a reliable, secure and efficient web application that provides all of these features for free to its users.

## 1.3 Scope of the Product

The photo editor project is a web application which can be hosted on any Computer. The application will be equipped with a user-friendly interface that allows users to easily enhance and manipulate their digital photos. The project will include features such as cropping, resizing, brightness/contrast adjustment, filters, PDF creation tool, mosaic creation, collage, noise removal and the ability to undo or redo actions.

The scope of the Photo Editor project includes the following features and functionalities:

- Editing tools: Crop, resize, noise removal, collage, mosaic, adjust brightness and contrast.
- Image format support: Support for various image formats, including JPEG, PNG.
- User-friendly interface: A clean and intuitive interface that makes it easy for users to navigate the software and apply editing tools.
- Output options: Ability to save images in different file formats and share them directly from the software.
- Additional support : Users also have the ability to enable/disable ‘high end passport creation flag ’ which allows users to use high end tools during passport creation or to use minimalistic features while generating passport images(generally used while creating a high number of images).

This scope represents the core functionality of the Photo Editor project, but additional features and improvements may be added over time based on user feedback and market demand.

# 2. Requirements

## 2.1 User Requirements

Basic hardware requirements such as :

- RAM – 4 GB
- The website should have an easy and user friendly interface to navigate between different available options.
- Users should be able to upload, view and edit the photos on the website and use the tool and edit them easily.
- The website should have different options such as resizing, format conversion, color adjustments, cropping, color correction, collage creation, noise removal, background change, mosaic creation, pdf creations, face detection, passport size creation tool.

- Users should be ensured about privacy of the data and that measures are in place to protect their photos.
- They should be able to download the edited image in their required format such as jpg, png, jpeg, etc.
- Users should be provided with high quality images without much pixel reduction or loss of quality.
- The website should load quickly and perform smoothly while handling complex or large size files.
- The website should be accessible to the users via web browsers and should be responsive enough in either case.
- This product is designed for photo editing and is accessible to all types of users.
- This product is being made available on the web and users should be comfortable accessing and using web-based applications on their computers.
- It is necessary for users to have either a basic understanding of uploading, editing, and downloading digital photos or some basic experience with using photo editing tools would suffice.
- Users are motivated to use our application for the purpose of effortlessly resizing and cropping their photos through the use of face detection technology, as well as creating passport size photos online without the need for a physical visit to a store.

## 2.2 Functional Requirements

The functionality of each and every feature is described in the home page and also the UI has different pages for each and individual features to process separately according to the user requirements and needs. Below is the list of functions available.

1. Passport photo maker
2. Background change
3. Resize
4. Crop
5. Noise removal
6. Image format conversion
7. Pdf maker
8. Photo collage
9. Photo mosaic
10. Brightness and contrast improvement

## 2.3 Non Functional Requirements

The system is able to handle ‘n’ number of users at a time without conflicts. Approximately it will take 2-5 seconds to process the passport creation and background change features. Remaining features will be very fast and will give immediate response.

When it comes to privacy, there won't be any storage of the user data. It will be auto deleted for every 10 mins of processing.

The system should be available 24/7, with an uptime of at least 99.9%.

## 2.4 System Requirements

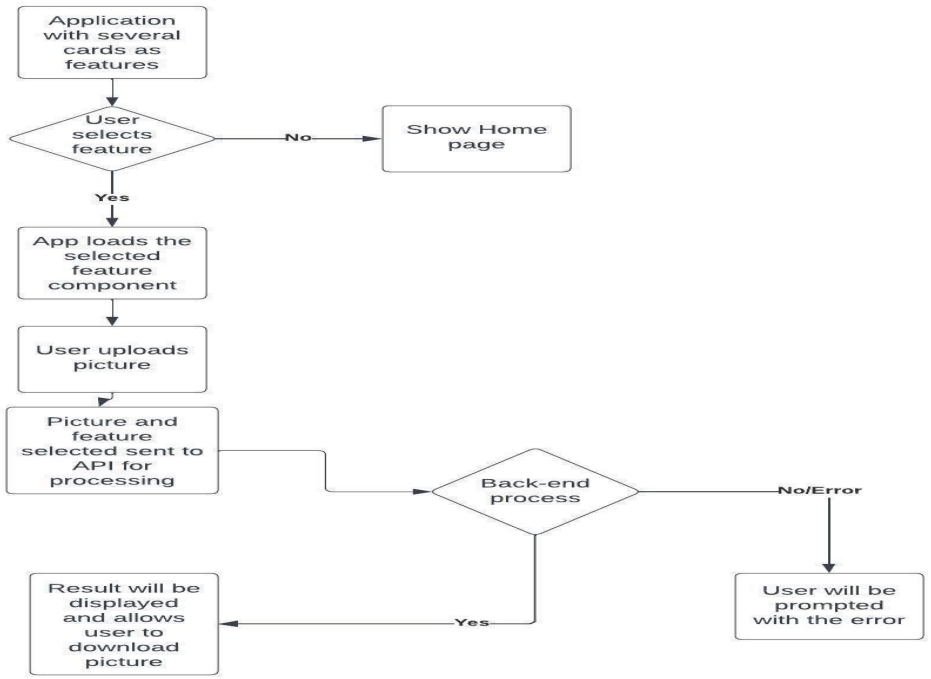
1. The website should run smoothly on standard hardware including laptops, desktops and tablets.
2. The website should run on all types of operating systems such as Linux, Mac OS and Windows.
3. It should be hosted on IBM cloud and required resources need to be assigned properly for the deployment of applications.
4. The website is developed using the modern programming languages like React Js, Python, Django and JavaScript.
5. The photo editing functionality is achieved using opencv libraries and respective image processing techniques.
6. The website should have better network connectivity to upload images and to ensure that they are processed properly.
7. The website should be optimised properly to perform image processing techniques and face detection algorithms to perform photo editing.
8. Proper logging, error and exception handling should be handled to accommodate any issues with the website.

## 2.5 Interface Requirements

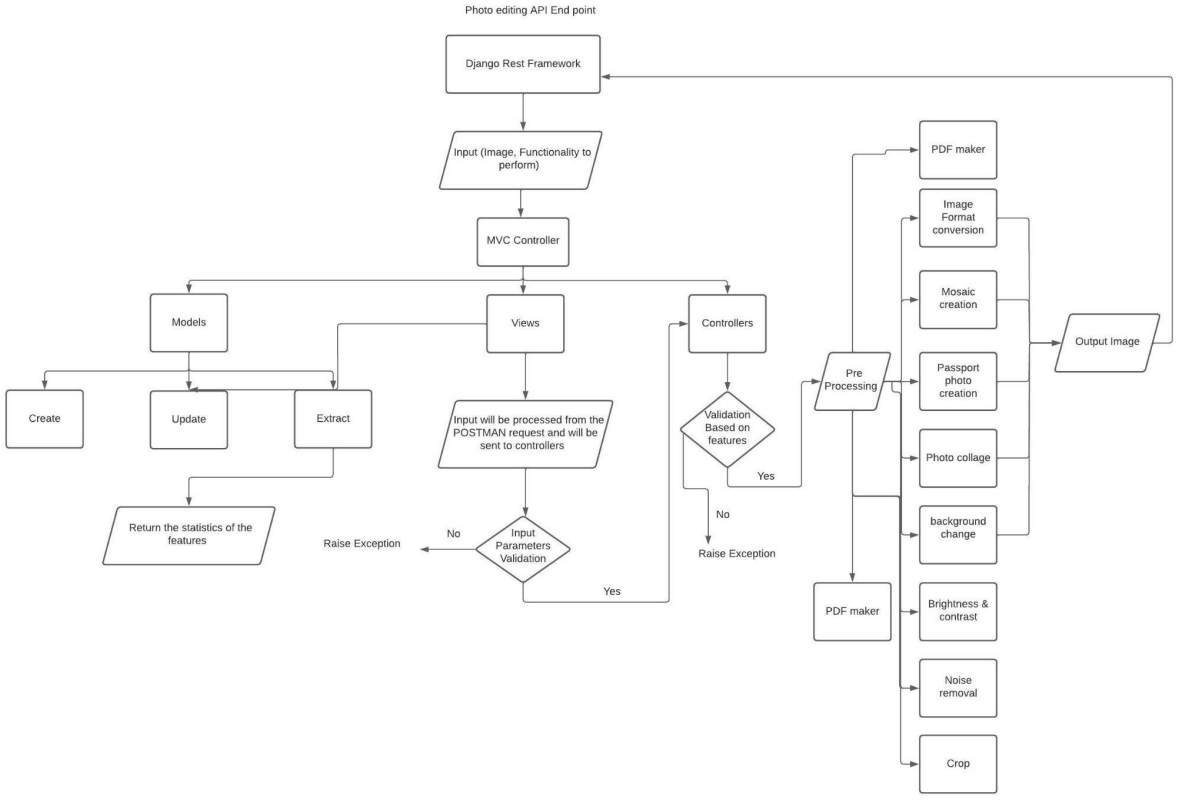
1. The Photo editing application is a Single Page Application (SPA) built by using React and Python as tech-stacks.
2. This SPA will have several cards with required features such as resize, crop, and image conversion and will also have a feature to download the image to your local.
3. Each card will have a feature title and a short description of what the feature does which helps the user to select.
4. Users can select any feature by selecting or clicking on the feature cards.
5. Upon selection, the component related to the selected feature will be loaded. Allowing users to use the feature he/she selected.
6. Toasters will be used to handle the errors in the interface.

# 3 System architectures and design

## 3.1 Front End




## 3.2 Backend




# Business Case for the Product

The Photo Editor product offers a solution to the increasing demand for high-quality images for personal or professional use. Key benefits include increased productivity, improved image quality, increased reach, increased revenue, and a competitive advantage. Though this product can be used by a wide variety of individuals it would help students with their passport size photos which is very crucial for job applications, presentations and so on.

### Product Functions

The photo editor is a comprehensive tool for enhancing and manipulating digital photos. The main functions that will be built into the product include:

- Cropping: Users will be able to crop their photos to remove unwanted elements or to focus on a specific area.
- Resizing: Users will be able to resize their photos to better fit their intended use, whether it's for social media, printing, or other purposes.
- Brightness/Contrast Adjustment: Users will be able to adjust the brightness and contrast of their photos to improve the overall look and feel.
- Upload/Download: As part of UI there will be a couple of buttons for Users to Upload images from their local system and download the same post image processing.
- Collage: Users will be able to collage multiple images into a set of collage formats displayed in the UI. Additionally to this Users will also be able to select which portion of the image to be included in the collage.
- Mosaic creation: Users will be able to create a Mosaic for any template such as UB logo and so on with a set of input images.
- Noise removal: Users will be able to upload any noisy image and get the noise removed from the same.
- Format Conversion: Users will be able to change the format of the input image to the desired versions such as png, jpg, jpeg and so on.
- PDF creation: Users will be able to merge multiple images into a single PDF file.
- Background change: Users will be able to change the background color of the image to White, Black or their desired colour of choice.

# 4. Implementation

## 4.1 Tech Stack

#### Front end

React js is used to build the front end UI interface. It allows us to use reusable components and

efficiently update DOM. HTML and CSS is used to style the UI pages and create the visual layout of the web pages. Javascript is used with the above entities and performs operations on the frontend and communicates with the APIs. Bootstrap is also used to make the website responsive. This provides the best pre-defined UI designs for components such as buttons, forms, navigation bars etc.

**Backend**

Python is used for the entire backend development for both features development and API building. Postgres will be used as a backend database. Apart from that there are various tools which will be used to develop the features are OpenCv, numpy, Pillow, Matplotlib, Pandas etc.

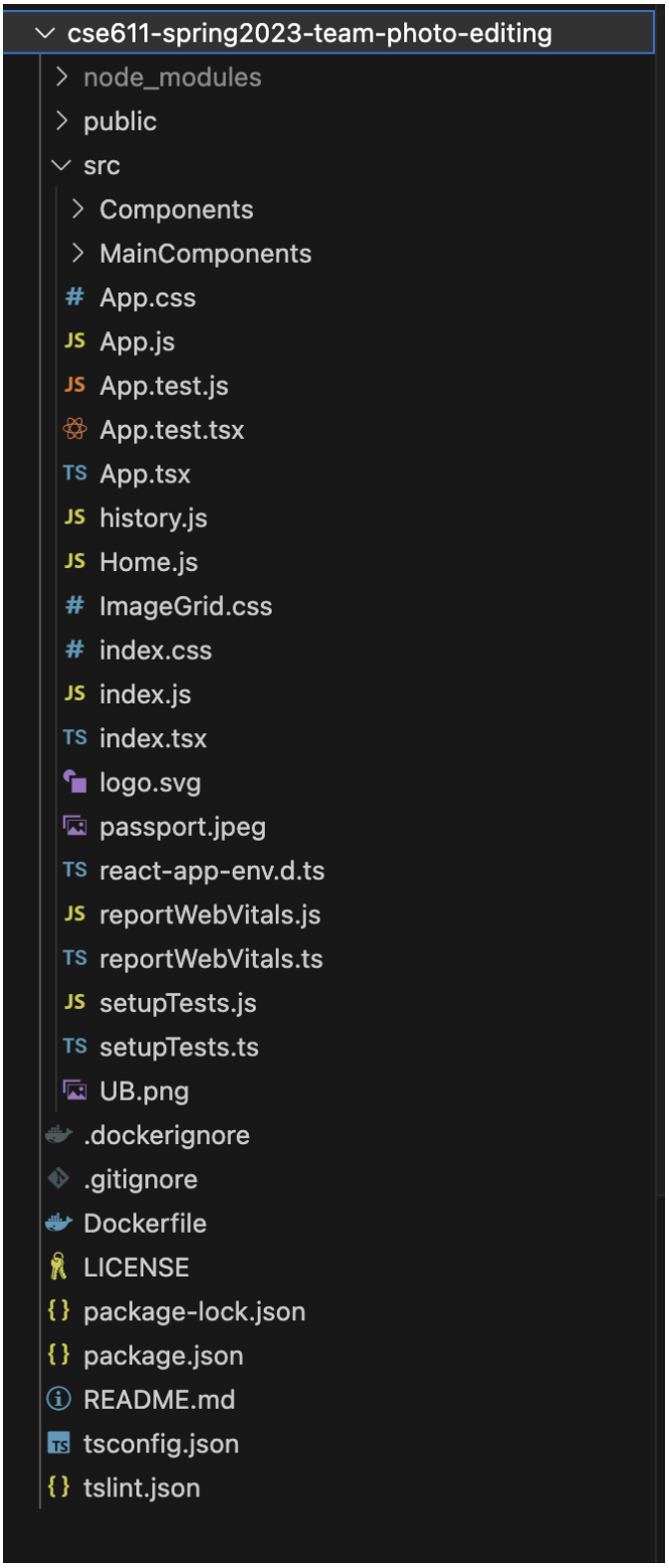
There are no specific system requirements for the application. Since it's a web application we just need a browser and a proper internet connection.

## 4.2 Implementation

## Front End

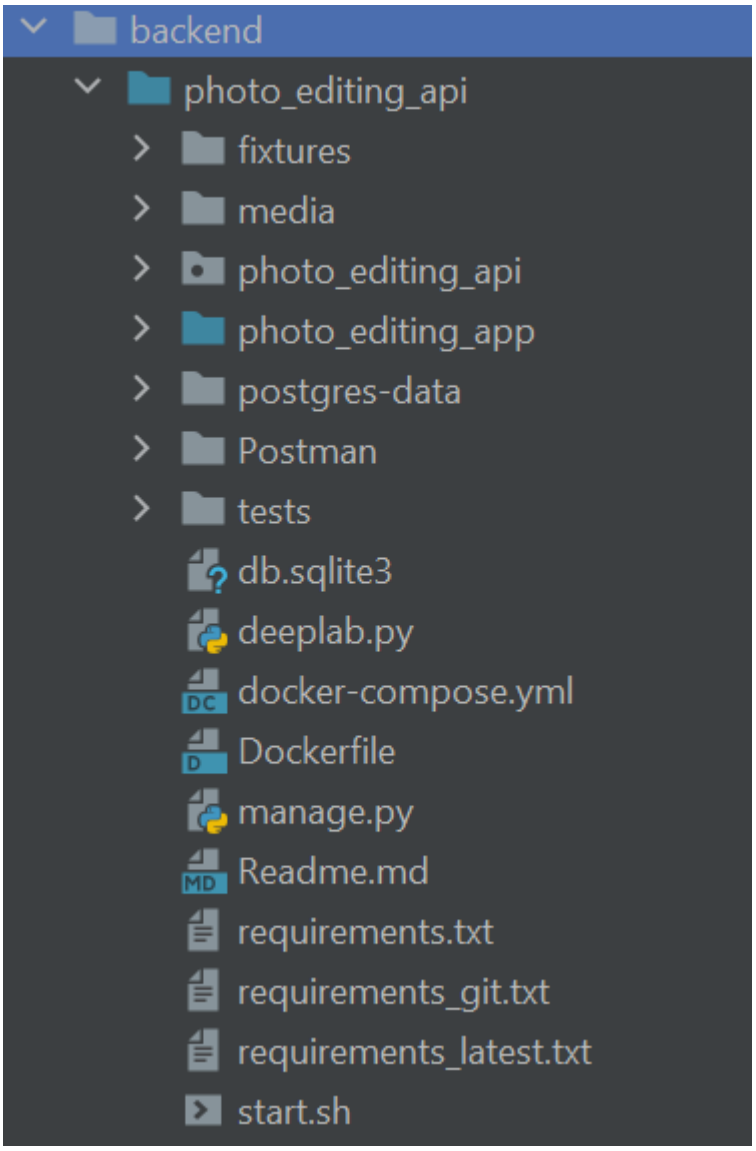
The frontend was completely developed using React.js, HTML, CSS, and JavaScript. It covers installation instructions, project structure, component architecture, state management, routing, and styling techniques. Additionally, it explores the integration of APIs, handling form data, and the usage of relevant frontend libraries and frameworks.

The below attached image is the entire project structure.
**Public**: Folder serves as the root directory for static assets that are not processed or transformed by the build system. It typically contains files like images, fonts, and external libraries that you want to include directly in your application without going through the build process.
**Components**: Folder consists of js files that are rendered in the landing page or home page of our application.
**MainComponents**: Folder typically contains feature components that represent the main functionality or sections of application. These components are often reusable and modular, encapsulating specific features or sections of user interface.
**Dockerfile**: Text file used to define the instructions for building a Docker image. It specifies the base image to use, any dependencies or packages to install, environment variables, and the commands needed to set up and run the application within a container.


cse611-spring2023-team-photo-editing
node_modules
public
src
Components
MainComponents
App.css
App.js
App.test.js
App.test.tsx
App.tsx
history.js
Home.js
ImageGrid.css
index.css
index.js
index.tsx
logo.svg
passport.jpeg
react-app-env.d.ts
reportWebVitals.js
reportWebVitals.ts
setupTests.js
setupTests.ts
UB.png
.dockerignore
.gitignore
Dockerfile
LICENSE
package-lock.json
package.json
README.md
tsconfig.json
tslint.json

## Back End

The backend was completely developed using Python, Django,Postgres, Opencv, Python Libraries etc.




The above is the entire project structure.

**Fixture** - This folder will contain the default database data.

**Media** - This folder contains the user data like input, output, models and datasets used for the application .

**Photo_editing_api** - This folder contains the django setting and urls and configurations related to django.

**Photo_editing_app** - This folder contains the controllers and all the code files related to each and every functionalities.

**Postman** - This folder contains the postman collections .

It will also contain the docker file, docker compose file requirements and readme files etc.

**Controllers**

This is main folder which contain all feature functionalities and

1. **Background_change -** This file contains the code to take the input as image and do operations related to background change and output the image with the same input image name.
2. **Cron1 -** This file is a cron job operations in django to delete the user data for every 10 mins
3. **Get_objects -** This file is used to get the data and update the data from the database.
4. **Mosaic_maker -** This file will take the template and some amount of images and perform the mosaic operations on it and output the mosaic image.
5. **Noise_removal -** This file is used to remove the noise from the image by taking the input image and performing some opencv operations using filters.
6. **Passport_photo -** This is the main feature of the application which is used for making a passport photo according to the countrywise, it was integrated with spec detector, pose detector, resize and background change functionalities.
7. **Pdf_maker -** This will create a pdf by taking the multiple images from the user and convert those as a pdf.
8. **Pose_detection -** This file contains the code for detecting the facial pose by detecting the eyes and then finding the centroids of both the eyes and drawing a line by connecting the 2 eyes centre points. Based on the line will find whether the face is tilted or not.
9. **Resize_operation -** This will take the width and height of the image as per the user requirements and process the image without affecting the pixels .
10. **Specs_finder -**
11. **Verify_params_controller -** This file will contain all the validations required for the API like validating whether the user is passing the parameters properly or not for a function and the format check of the input files etc.

# 5. Testing

### 5.1 Functionality Testing

We have done Functional testing for our application which mainly focuses on verifying whether our application performs its functions correctly and meets the specified requirements. The main goal of functional testing is to ensure that the software meets the end-users' needs and works as intended, without any critical defects or issues.

**Passport Photo Feature:**

| **Functions/Scenarios (Inputs and Overall Functionality)** | **Expected Behaviour (Output)** | **Actual Result of the Application.** |
|---|---|---|
| User uploads an image and selects the country and the passport photo creation tool needs to give a digital image which satisfies all the passport photo requirements of that specific country selected. | The tool needs to give a digital image which satisfies the country requirements like background change to a light colour. Applies border to the images and resizes the image based on the country requirement. | **Working as Expected** |
| User uploads an image and selects the country: France and the passport photo creation tool needs to give a digital image which satisfies all the passport photo requirements of that specific country selected. | The tool needs to validate whether the user is wearing spectacles or not. In UI we need to throw a pop-up to the user saying spectacles are not allowed for this country, if the image uploaded contains spectacles. | **Working as Expected** |
| User uploads an image and selects any country of choice and the passport photo creation tool needs to give a digital image which satisfies all the passport photo requirements of that specific country selected. | The tool needs to perform face and pose detection for any image that is uploaded.<br>If there is no face in the images (like any missing facial features like eyes, nose, lips, chin or cheeks) throw a pop-up to the user saying face is not detected.<br>If the image has any left or right tilt, throw a pop-up to the user saying the image is tilted and they need to re-upload the image. | **Working as Expected** |

**Background Change Feature:**

| **Functions/Scenarios (Inputs and Overall Functionality)** | **Expected Behaviour (Output)** | **Actual Result of the Application.** |
|---|---|---|
| The user uploads an image and chooses a particular colour, and then the background change tool should provide a digital image with a background that matches the selected colour. | The tool needs to change the background of the image to the selected colour. | **Working as Expected** |

**Crop Feature:**

| **Functions/Scenarios (Inputs and Overall Functionality)** | **Expected Behaviour (Output)** | **Actual Result of the Application.** |
|---|---|---|
| The user uploads an image and chooses a particular image ratio, Then the crop tool should provide a digital image with the desired part. | The tool should eliminate any undesired elements in the image and only provide the selected portion to the user. | **Working as Expected** |

**Collage Feature:**

| **Functions/Scenarios (Inputs and Overall Functionality)** | **Expected Behaviour (Output)** | **Actual Result of the Application.** |
|---|---|---|
| Users choose a template, and Based on the template chosen users need to upload multiple images and expect an output image that combines all of the input images. | The tool must offer various templates for varying numbers of images and produce a single image by merging multiple images according to the selected template. | **Working as Expected** |

**Noise Removal Feature:**

| **Functions/Scenarios (Inputs and Overall Functionality)** | **Expected Behaviour (Output)** | **Actual Result of the Application.** |
|---|---|---|
| User chooses a digital image and we need to have an image without noise. | The tool must provide sharp and clean images by removing any unwanted distortions or noise in the image. | **Working as Expected** |

**Format Conversion Feature:**

| **Functions/Scenarios (Inputs and Overall Functionality)** | **Expected Behaviour (Output)** | **Actual Result of the Application.** |
|---|---|---|
| Users upload a digital image and choose an image format and the tool must return the image in the needed format . | The tool must accept any image format and needs to change the file format of the image to JPG or PNG or JPEG and also we preserve the visual content of the image. | **Working as Expected** |

**Brightness & Contrast Feature:**

| **Functions/Scenarios (Inputs and Overall Functionality)** | **Expected Behaviour (Output)** | **Actual Result of the Application.** |
|---|---|---|
| Users upload an image. They should be able to change the brightness and contrast of the image dynamically. | The tool should be capable of adjusting the brightness and contrast of an image. This means that the user should have the ability to make the image either brighter or darker, as well as increase or decrease the contrast between light and dark areas within the image. | **Working as Expected** |

**Resize Feature:**

| **Functions/Scenarios (Inputs and Overall Functionality)** | **Expected Behaviour (Output)** | **Actual Result of the Application.** |
|---|---|---|
| Users upload an image and provide width and height, the output image should be resized to needed specifications. | The tool must have the capability to modify the width and height of an image as per user input, and generate a resized image that matches the specified width and height parameters. | **Working as Expected** |

**PDF Maker Feature:**

| **Functions/Scenarios (Inputs and Overall Functionality)** | **Expected Behaviour (Output)** | **Actual Result of the Application.** |
|---|---|---|
| Users upload multiple images and should provide a pdf with all images | The tool should have the ability to merge multiple images into a PDF format while maintaining the original image size. | **Working as Expected** |

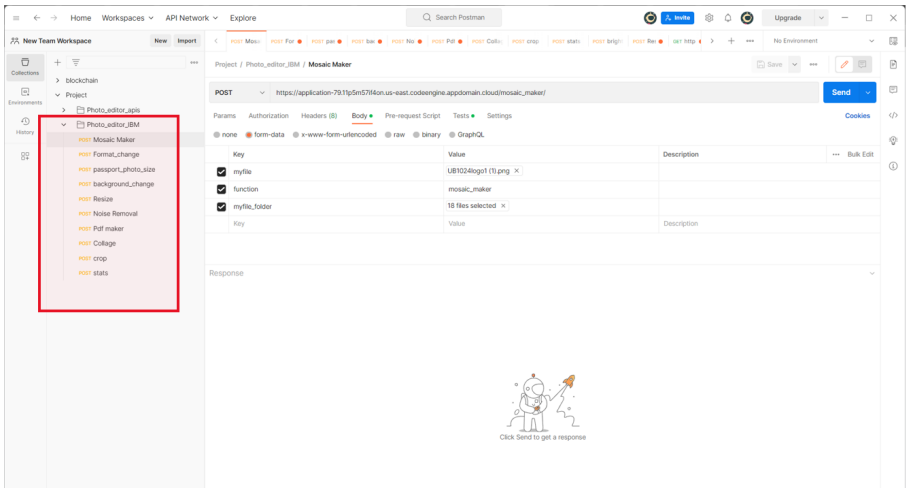
**Mosaic Feature:**

| **Functions/Scenarios (Inputs and Overall Functionality)** | **Expected Behaviour (Output)** | **Actual Result of the Application.** |
|---|---|---|
| Users upload a template image and multiple tile images. the tile images should be arranged in the shape of the template image | The tool is expected to replace the template image with the required tile images while preserving the shape of the original template image. | **Working as Expected** |

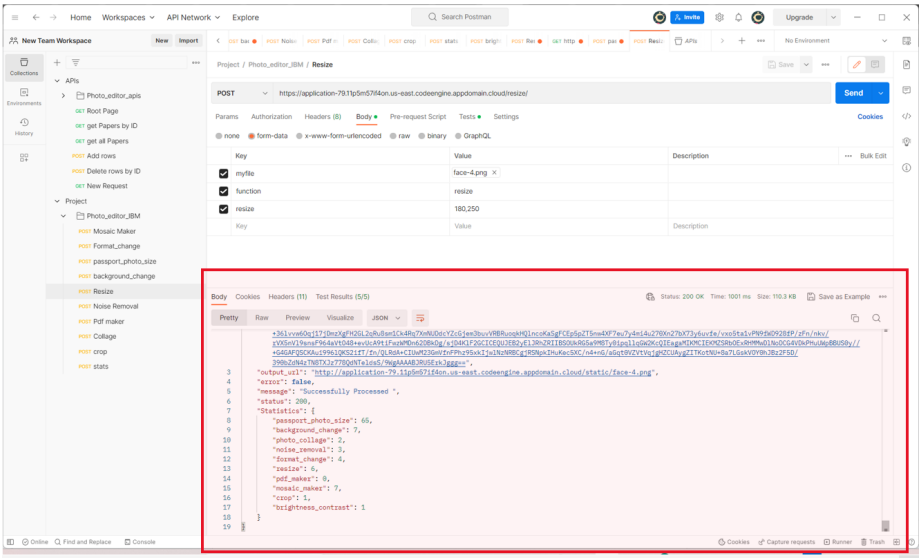
### 5.3 API testing

1. For API testing you need to import a collection file in Postman, you can find the collection file here.

2. After import you should be able to see below collection in your postman

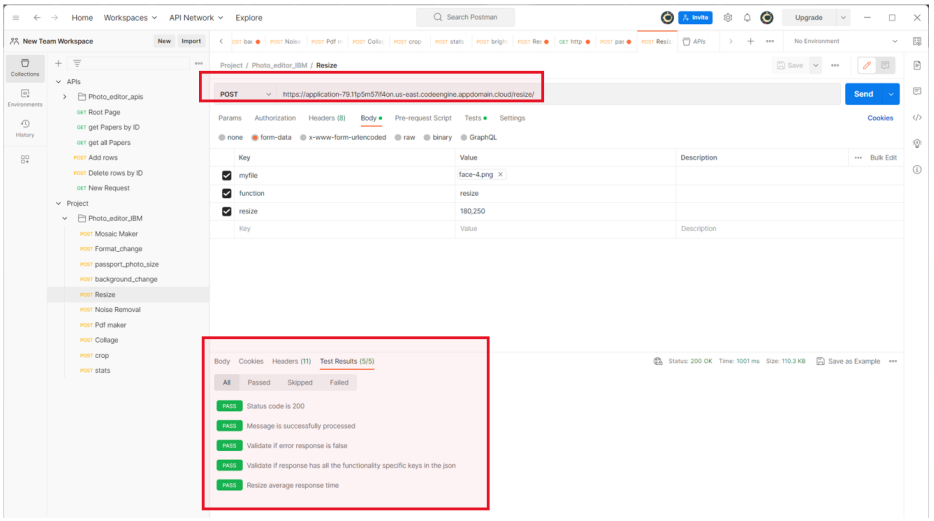



3. On successfully running any API you should be able to view results as below




We have added test scripts as part of the API, so whenever you run any API then in the background these tests will be executed.

Below are the test results for resize API.




## APIs information:

1. Mosaic
   https://application-79.11p5m57if4on.us-east.codeengine.appdomain.cloud/mosaic_maker/ : Creates a Mosaic out of input images based on the template selected.
   Input parameters of the API are as below:
   'myfile': 'imagefilepath',
   'function': 'mosaic',
   'myfile_folder': 'multipleimagesfilepaths'

2. Passport-creation
   https://application-79.11p5m57if4on.us-east.codeengine.appdomain.cloud/passport_photo_size/ : Creates passport size photo along with other optional constraints such as specs required or not, minimalistic features applied on pictures or not.
   Input parameters of the API are as below:
   'myfile': 'imagefilepath',
   'function': 'passport_photo_size',
   'country':'countryname'
   'background_req':'yes/no'

3. Format-change
   https://application-79.11p5m57if4on.us-east.codeengine.appdomain.cloud/format_change/ : Changes the image format based on user selection
   Input parameters of the API are as below:
   'myfile': 'imagefilepath',
   'function': 'format_change',

4. Background-change
   https://application-79.11p5m57if4on.us-east.codeengine.appdomain.cloud/background_change/ : Changes the background colour of the image to the one selected by user.
   Input parameters of the API are as below:
   'myfile': 'imagefilepath',
   'function': 'background_change',

5. Resize
   https://application-79.11p5m57if4on.us-east.codeengine.appdomain.cloud/resize/:
   Changes the dimensions of the images based on the user input.
   Input parameters of the API are as below:
   'myfile': 'imagefilepath',
   'function': 'resize',

6. Noise-removal
   https://application-79.11p5m57if4on.us-east.codeengine.appdomain.cloud/noise_removal/ : Removes noise from the given input image.
   Input parameters of the API are as below:
   'myfile': 'imagefilepath',
   'function': 'noise_removal'

7. Pdf-maker -
   https://application-79.11p5m57if4on.us-east.codeengine.appdomain.cloud/pdf_maker/
   : Collages input images to a PDF file.
   Input parameters of the API are as below:
   'myfile': 'imagefilepath',
   'function': 'pdf_maker'

8. Statistics
   https://application-79.11p5m57if4on.us-east.codeengine.appdomain.cloud/stats/
   : Returns statistical information such as API hit count for each API.
   Input parameters of the API are as below:
   'function': 'stats'

# 6. Deployment

### Front End

***Local Environment setup***

1. Git clone using the command
   ***git clone*** ***https://github.com/xlab-classes/cse611-spring2023-team-photo-editing.git***
2. Navigate to cse611-spring2023-team-photo-editing
   ***cd*** *cse611-spring2023-team-photo-editing*
3. Install node modules using " npm install " or " npm install –force".
4. Start your application using " npm start ", you can view application running on port 3000.

***Docker Environment setup***

1. Build image using **docker buildx build --platform linux/amd64 --output type=docker .**
2. Check your images using **docker images -a**
3. Now tag the latest image using **docker tag <ImageID> arjun3816/photoeditor_ui:0.3** ( in our case ).
4. Push your changes to docker using **docker push TAGNAME:VERSION**
5. Open IBMCloud and click on save and edit configuration and then save.
6. Now you can test application using **Test application.**

### Back End

***Local Environment setup***

1. Git clone using the command
   ***git clone*** ***https://github.com/xlab-classes/cse611-spring2023-team-photo-editing.git***
2. Create a python environment variable
3. Activate the ENV and install all the requirement using the
   ***pip install -r requirements.txt***
4. Change the database configuration in the **settings.py**
5. Once the database has been created run the following commands.
   ***Python manage.py runserver migrate***
   ***Python manage.py runserver makemigrations***
   ***Python manage.py runserver loaddata fixtures/data.json***
6. Finally run the **python manage.py runserver**

***Docker Environment setup***

The following are the steps to deploy the new version of the existing project environment in docker

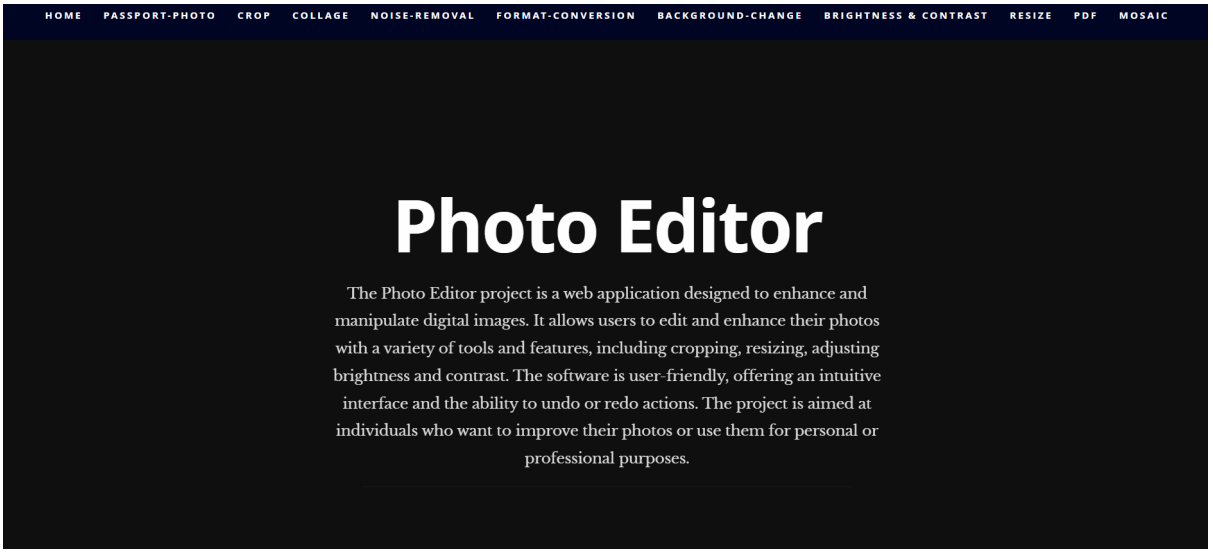
- Build the image using the **docker build -t tagname:version**
- Once the image is build then up the images using the **docker-compose up**
- The environment will be set automatically in the image and by using the **start.sh** all the commands to run will be up automatically.

***IBM Cloud Setup***

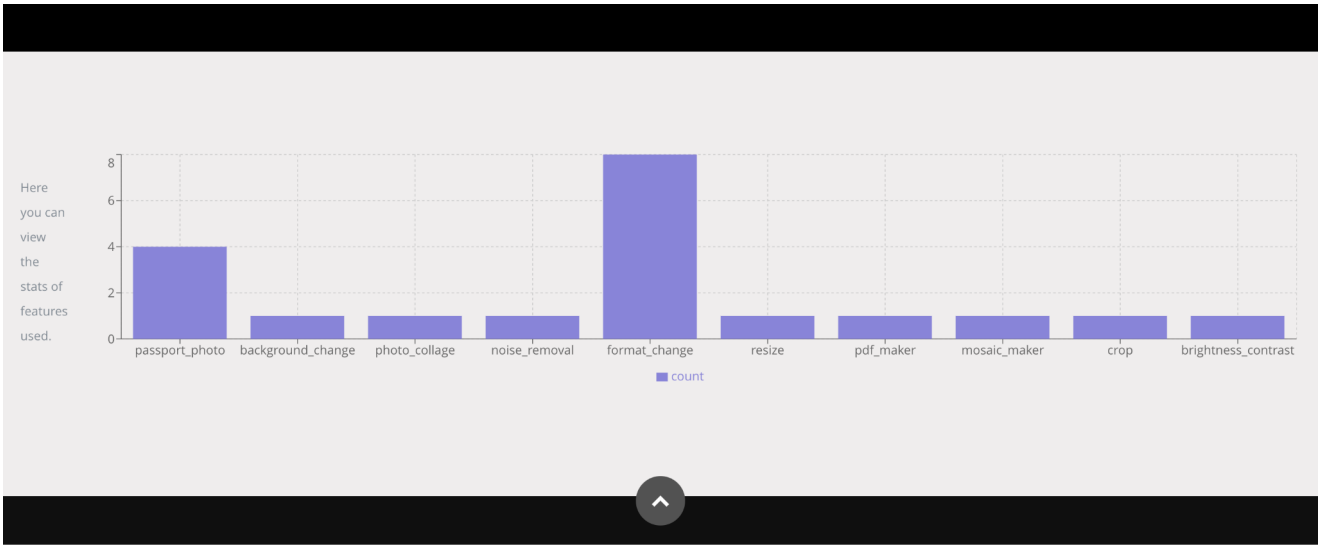
- Once the image is built then navigate to the **code Engine/ project/application.**
- Create an application and configure the docker image name of the latest version and it will create the instance and it will be ready to use.
- Front End - https://photoeditorui.11p5m57if4on.us-east.codeengine.appdomain.cloud/
- BackEnd - https://application-79.11p5m57if4on.us-east.codeengine.appdomain.cloud/
- The above 2 urls can be used for testing.

# 7.Handbook

The below image is the home page of our application which is shown as we enter the website url in any browser.



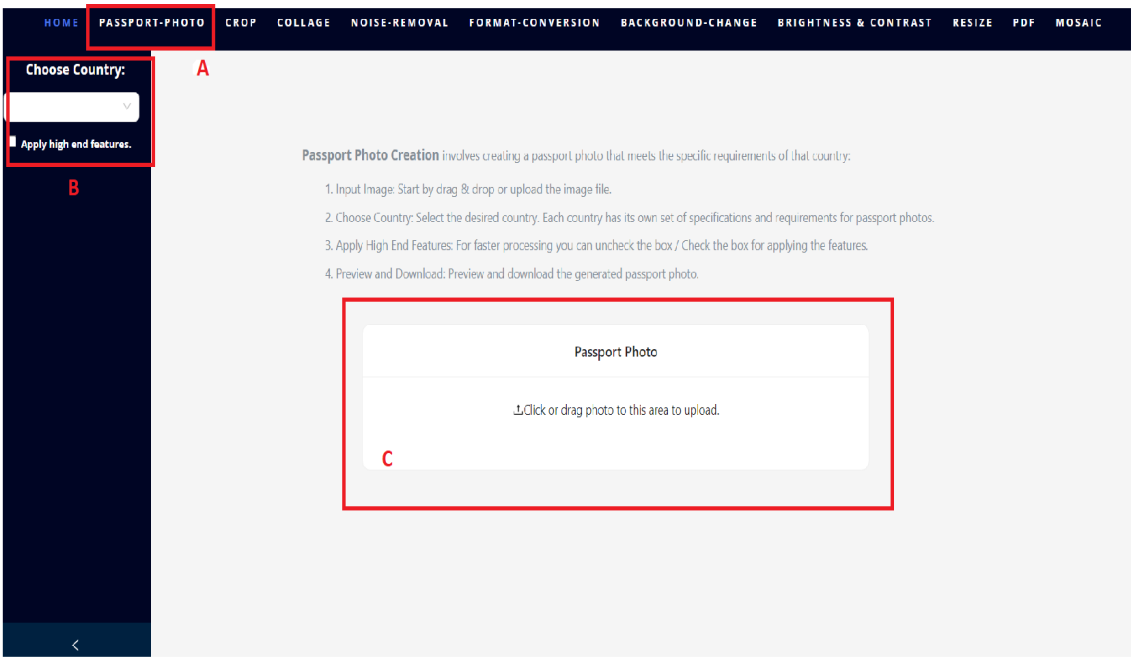

It has all the buttons which are used to navigate to different features.

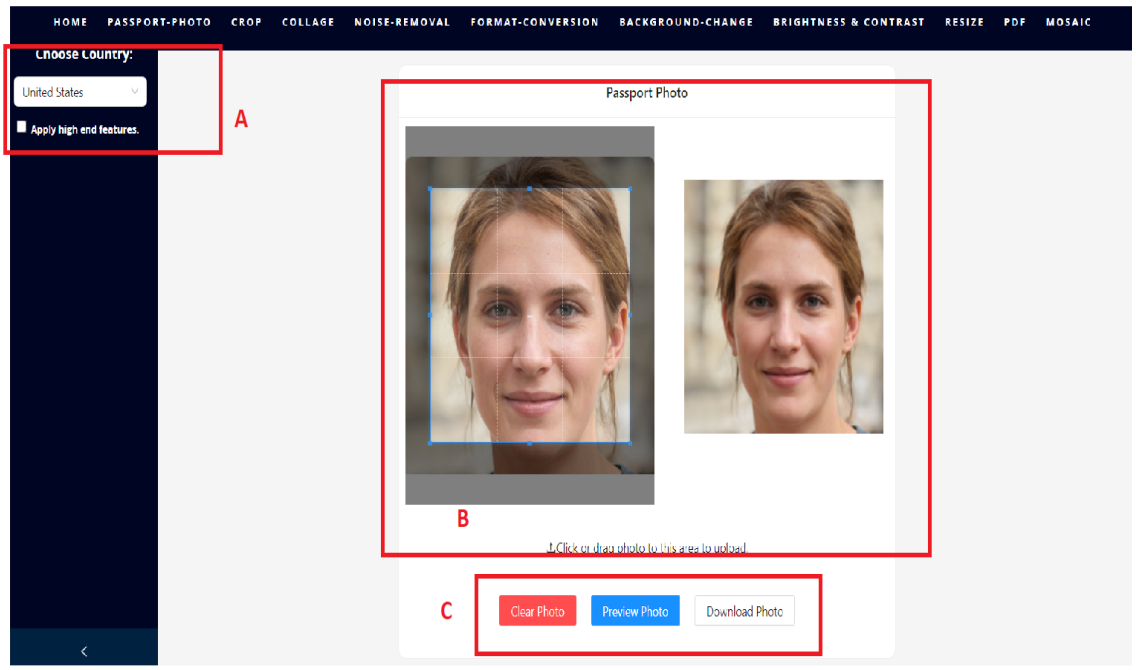



The above image displays the statistics of how widely the features are used, giving an overview of all available features and its usage count in the form of pixels.

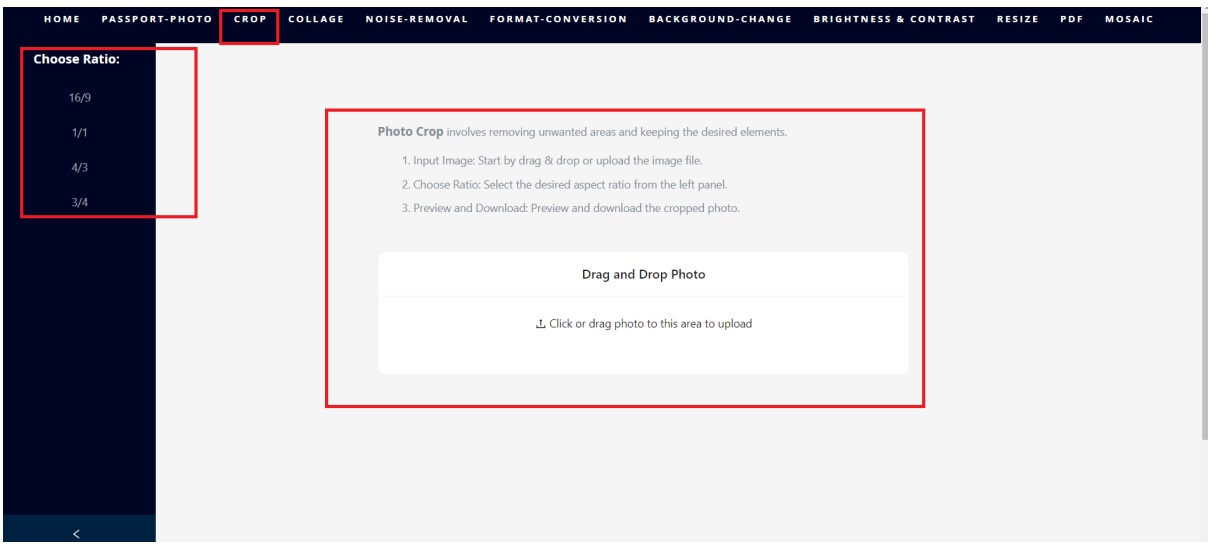
**Passport-Port:**

Now when passport-photo is clicked (A) , the user can select a country from the drop down box in (B) and also check or uncheck a checkbox present which applies all high end features required for that country's passport photo creation. Region (C) shows the description of what this feature does and an area to upload images on which this should be applied.

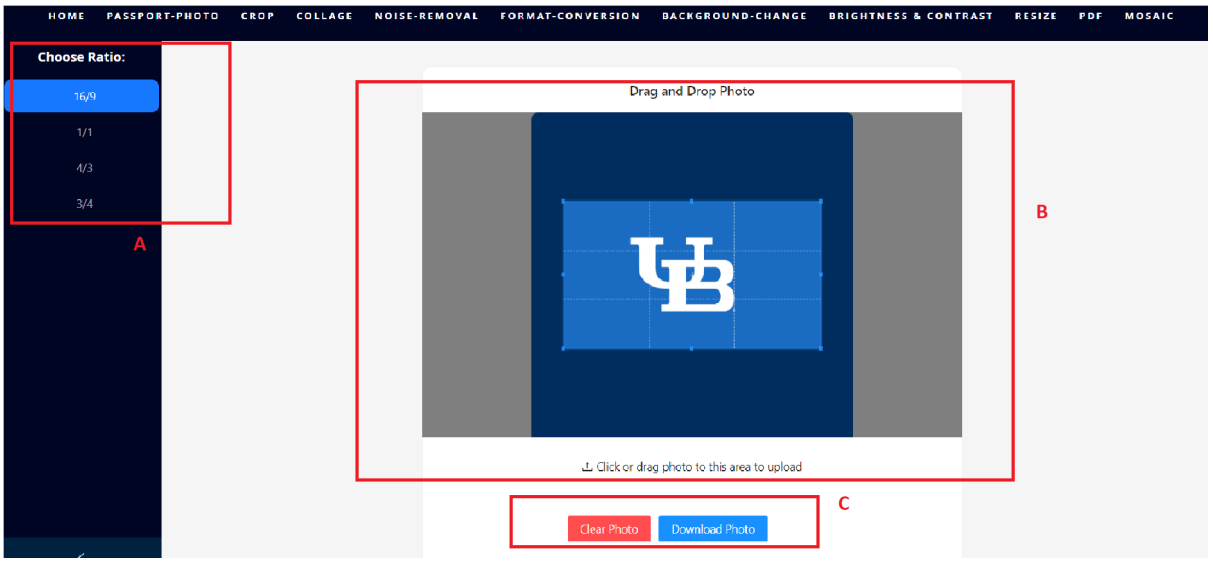



HOME
PASSPORT-PHOTO
CROP
COLLAGE
NOISE-REMOVAL
FORMAT-CONVERSION
BACKGROUND-CHANGE
BRIGHTNESS & CONTRAST
RESIZE
PDF
MOSAIC
Choose Country:
United States
Apply high end features.
A
Passport Photo
B
Click or drag photo to this area to upload
C
Clear Photo
Preview Photo
Download Photo

## Crop:


HOME
PASSPORT-PHOTO
CROP
COLLAGE
NOISE-REMOVAL
FORMAT-CONVERSION
BACKGROUND-CHANGE
BRIGHTNESS & CONTRAST
RESIZE
PDF
MOSAIC
Choose Ratio:
16/9
1/1
4/3
3/4
Photo Crop involves removing unwanted areas and keeping the desired elements.
1. Input Image: Start by drag & drop or upload the image file.
2. Choose Ratio: Select the desired aspect ratio from the left panel.
3. Preview and Download: Preview and download the cropped photo.
Drag and Drop Photo
Click or drag photo to this area to upload

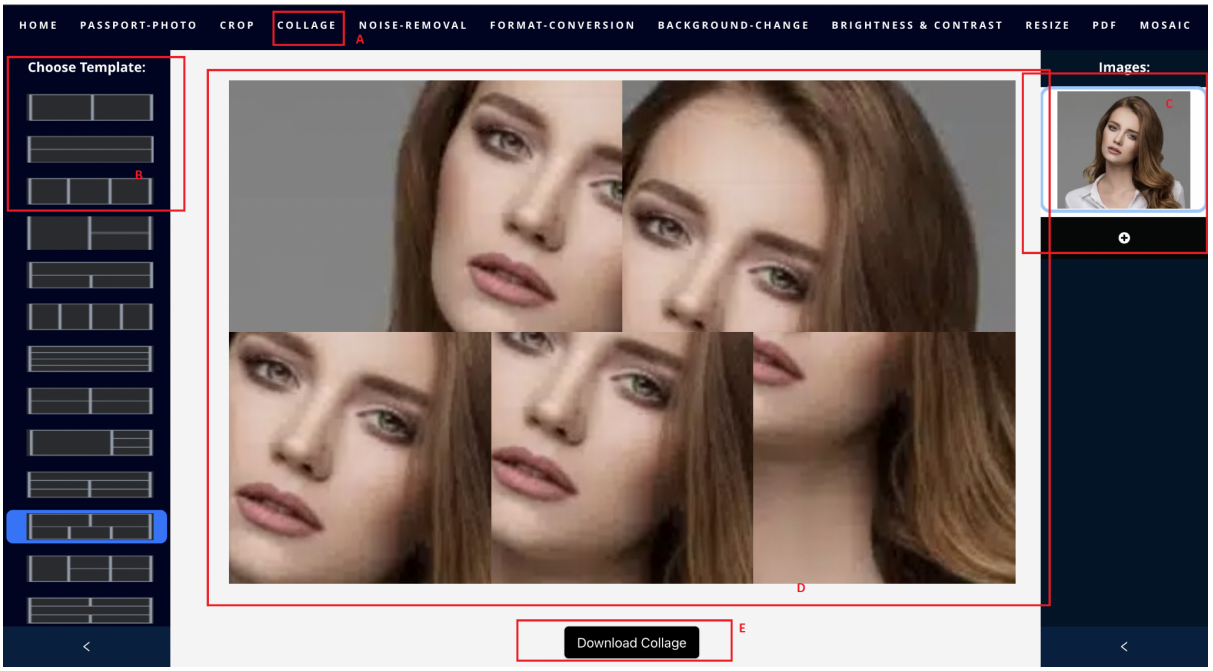
The above image shows that aspect ratio can be selected once clicked on the crop feature in navbar and then a description is provided and an area to upload image that needs to be cropped.




Region (A) is to select aspect ratio for cropping, region (B) is to display image and Region (C)is to either clear uploaded image or download image that is cropped.

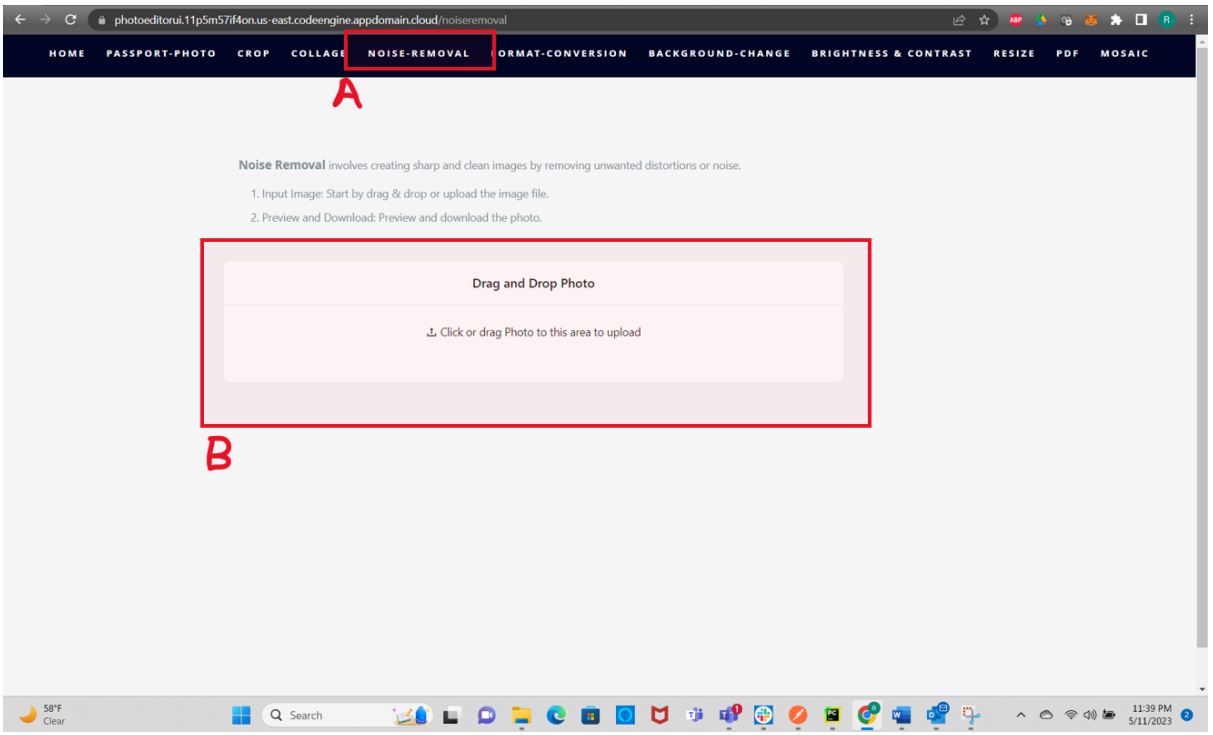
**Photo-Collage:**

Next click on collage to perform a collage feature on your uploaded images. Users select a template from region (B) and upload images to be collaged in region (C) and once upload is done, drag and drop the images in respective areas in region (D) and then zoom in/zoom out the images (also can remove the uploaded image). Once everything looks good click on download to download the collage.

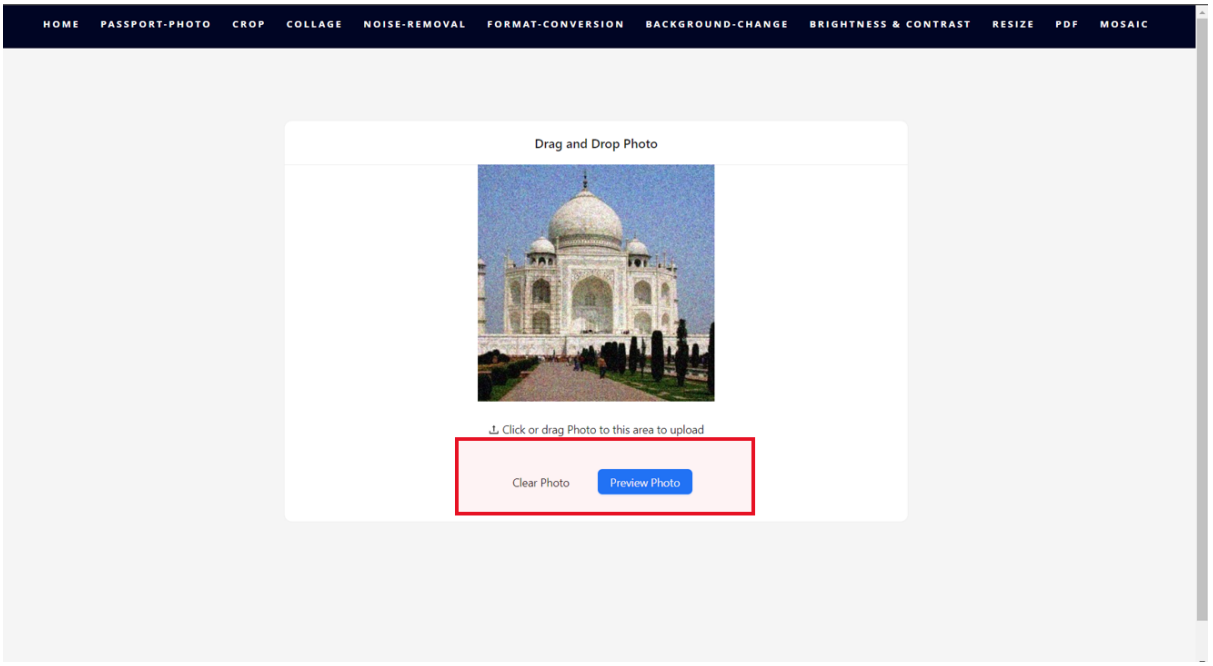



**Noise Removal:**

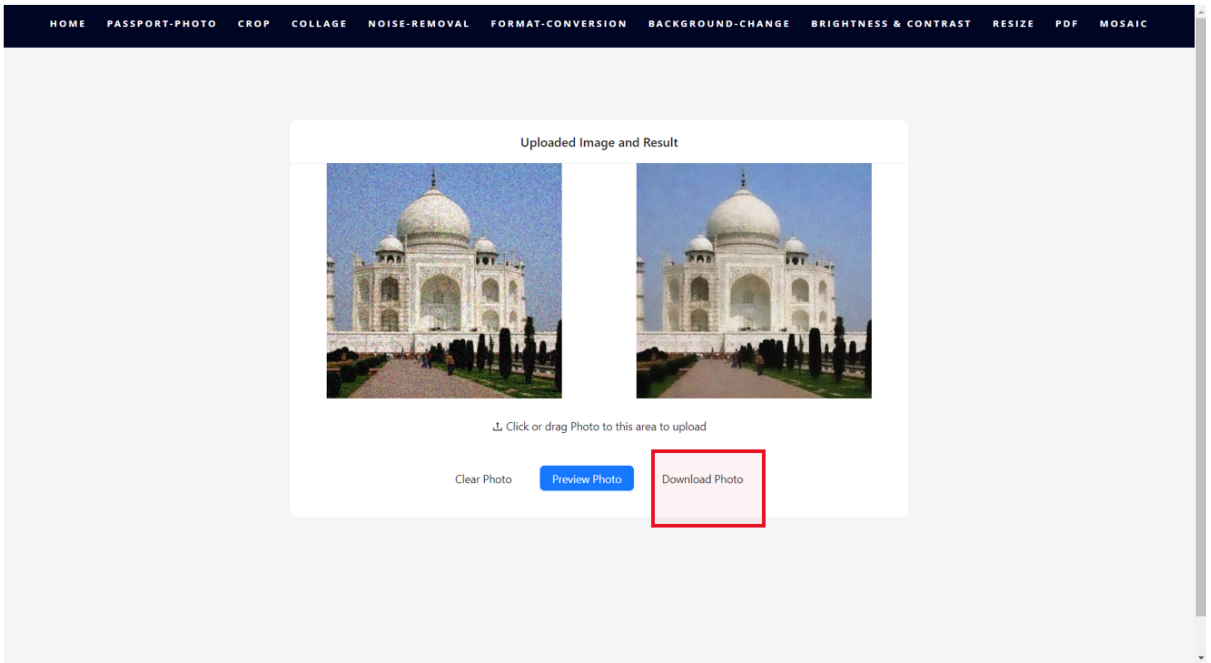
For noise removal please click on the ‘NOISE-REMOVAL’ button(region A) as below. To upload an image, the user can use region B and he/she can either drag & drop or they can browse the image as shown in the below figure.




After the image gets uploaded users will be able to view the below. Now users can click on ‘Clear Photo’ button if they want to reload the image or click on ‘Preview button’ to view the cleaned image (image we get after removing noise).

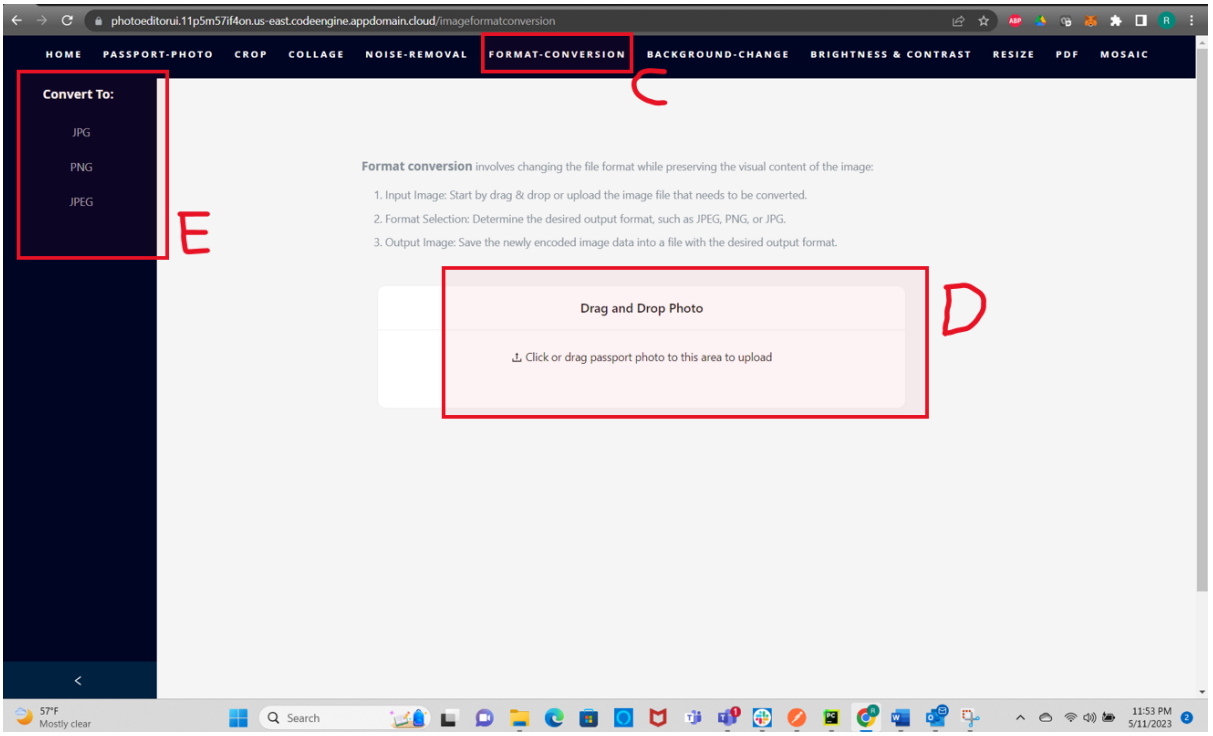



Below is the image we get after noise removal. If the user is satisfied he/she can download the image by clicking on the ‘Download button’.

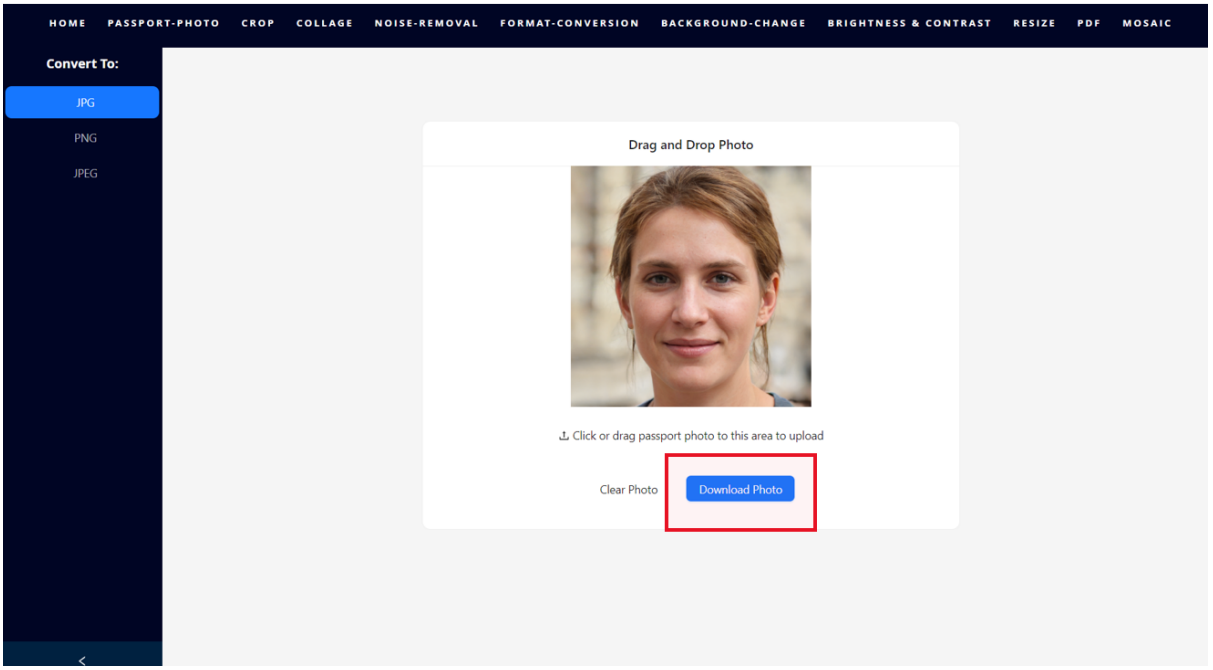



**Format Conversion :**

For Format conversion please click on the ‘FORMAT-CONVERSION’ button(region C) as below. To upload an image, the user can use region D and he/she can either drag & drop or they can browse the image as below. Users can select any of the desired format as shown in the region E.

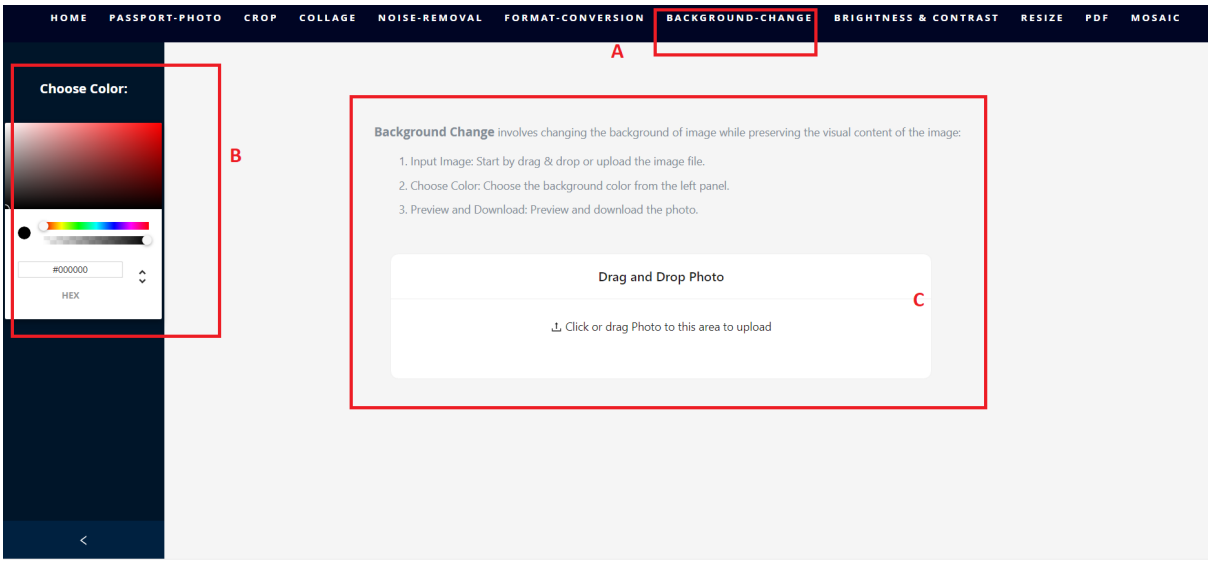



After the image gets uploaded users will be able to view the below. Now users can click on ‘Clear Photo’ button if they want to reload the image or click on Download button’ to view the formatted image as shown in the below figure.

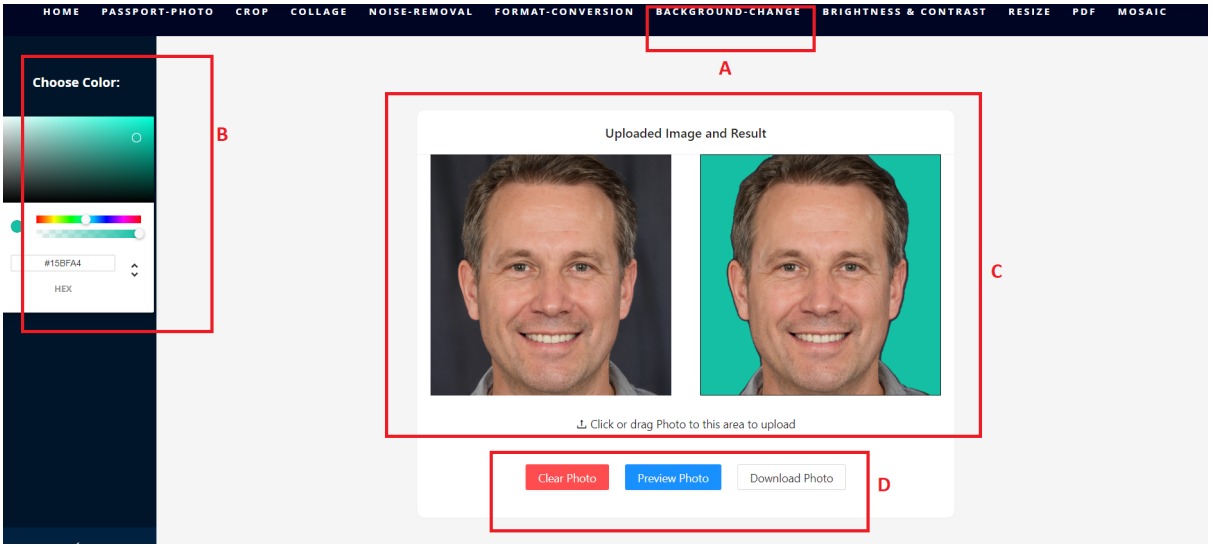



**Background Image Change:**

Click on Region (A) which is Background-Change and then select colour to which background needs to be changed from region (B) and Region (C)has the description on how the feature works and an area where the image can be uploaded.

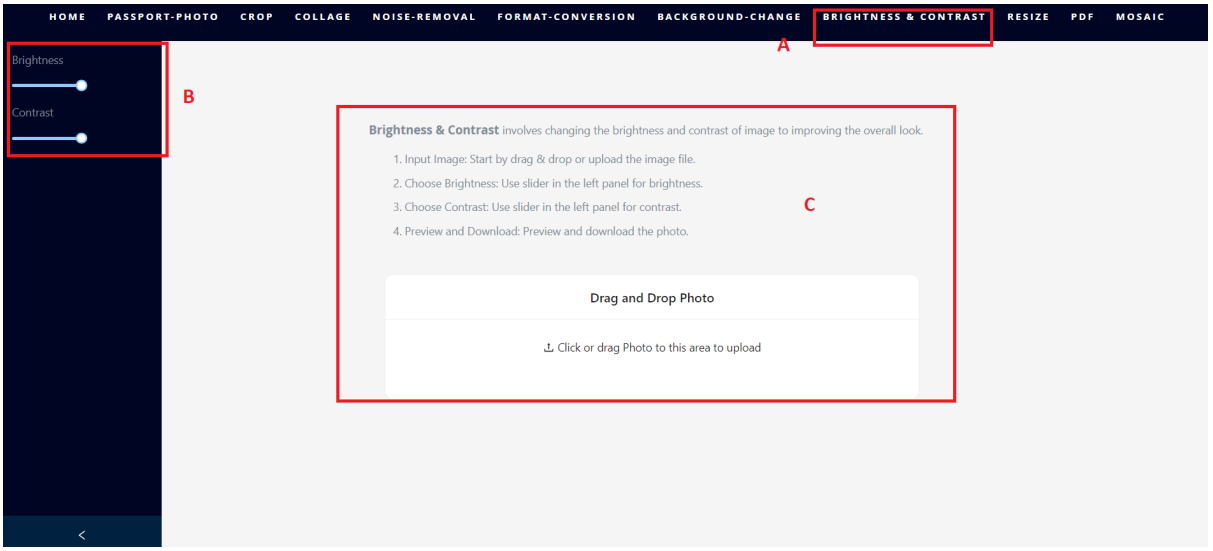



Once the above steps are done, the result/ preview is displayed as shown in Region (C)here in below image with the original image and once everything looks good, click on Download in Region (D) and the image is downloaded with new background.

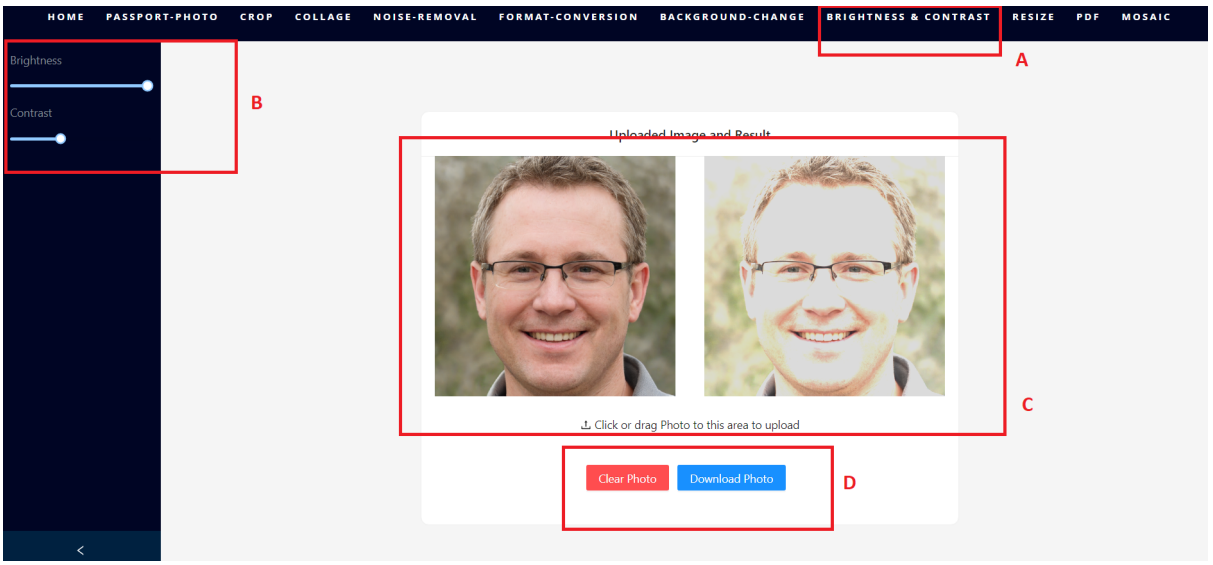



**Brightness & Contrast :**

Next feature is Brightness & Contrast when clicked by the user can update the brightness and contrast values on a slider in Region (B) and Region (C) has description on how the feature works and an area to upload the image.

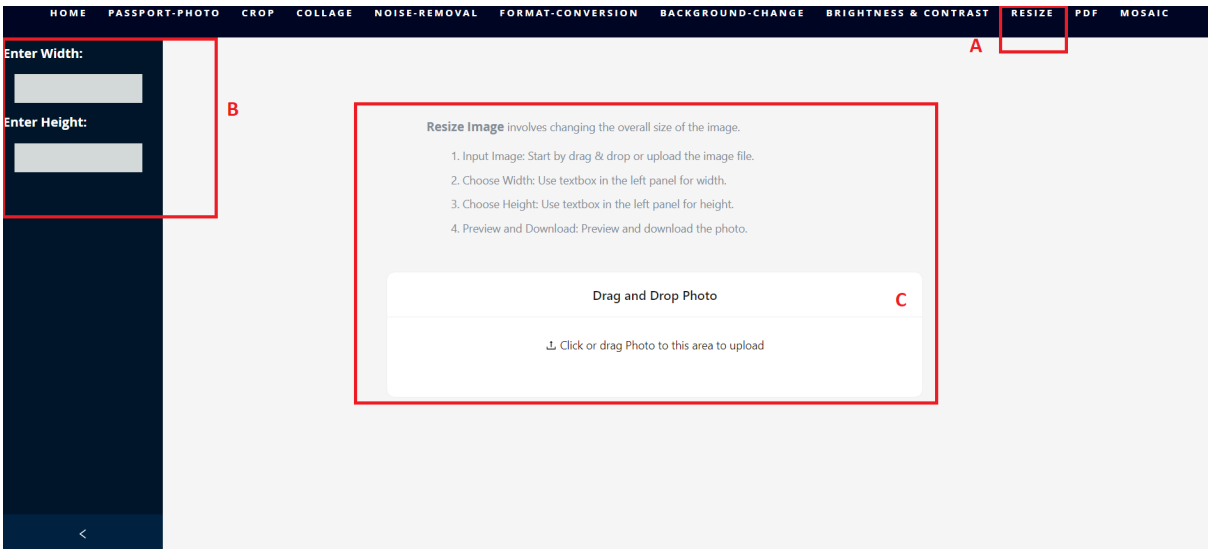



As shown in the image below, the original image and updated image is displayed in Region (C) and user can download the new image by clicking on Download button in Region (D).

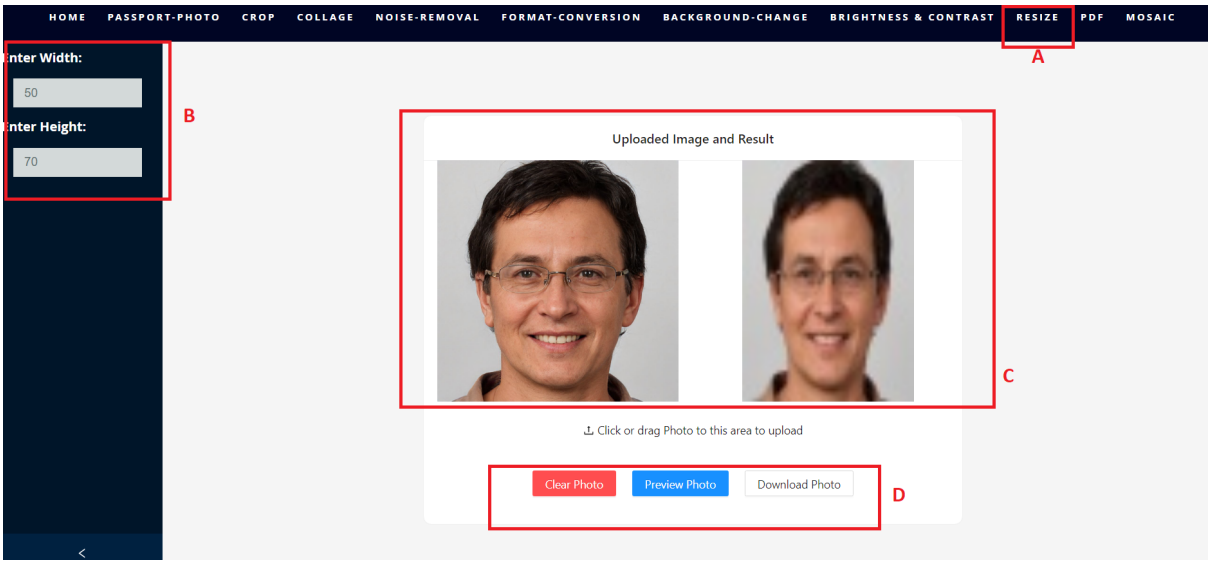



**Resize :**

Next feature is Resize when clicked by the user can change the size in input boxes in Region (B) and Region (C) has description on how the feature works and an area to upload the image.

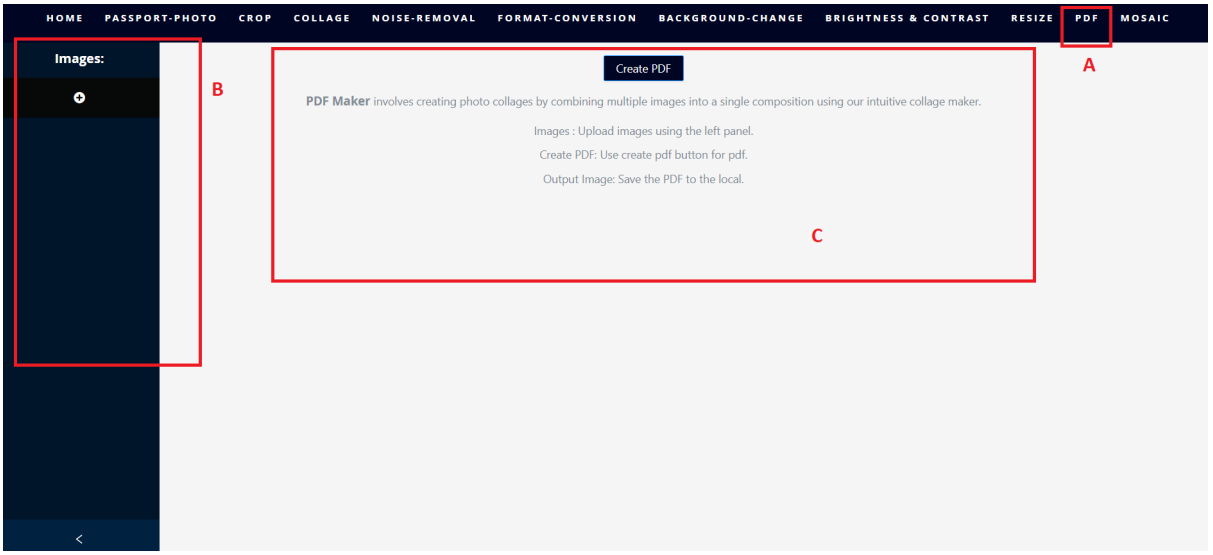



As shown in the image below, the original image and updated image is displayed in Region (C) once preview is clicked, and users can download the new image by clicking on Download button in Region (D).

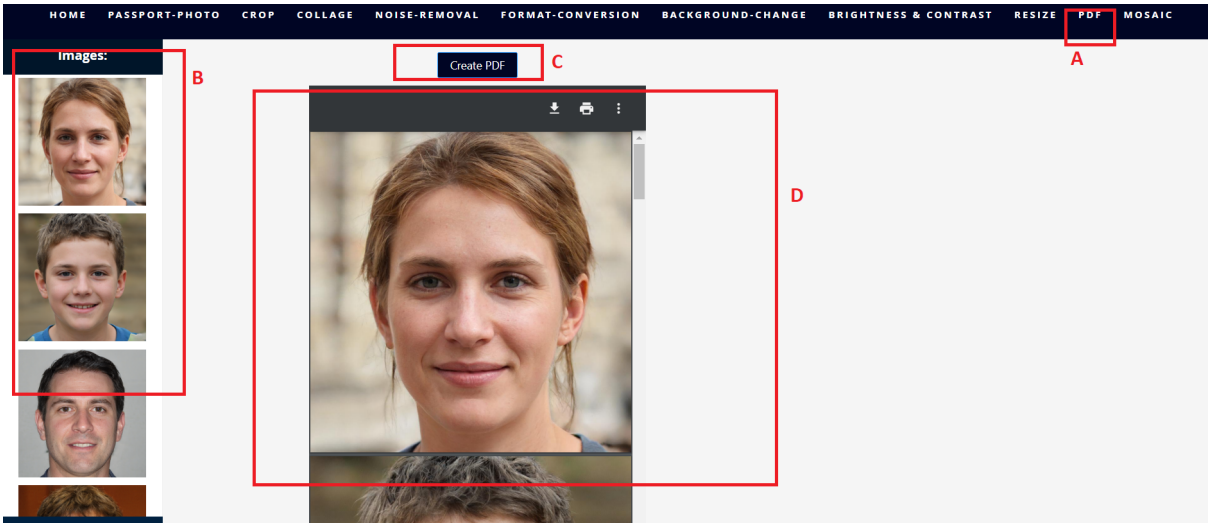



**PDF Creator:**

Next feature is PDF Creator when clicked by the user can create PDF out of uploaded images in Region (B) and Region (C) has description on how the feature works and an area to display PDF.

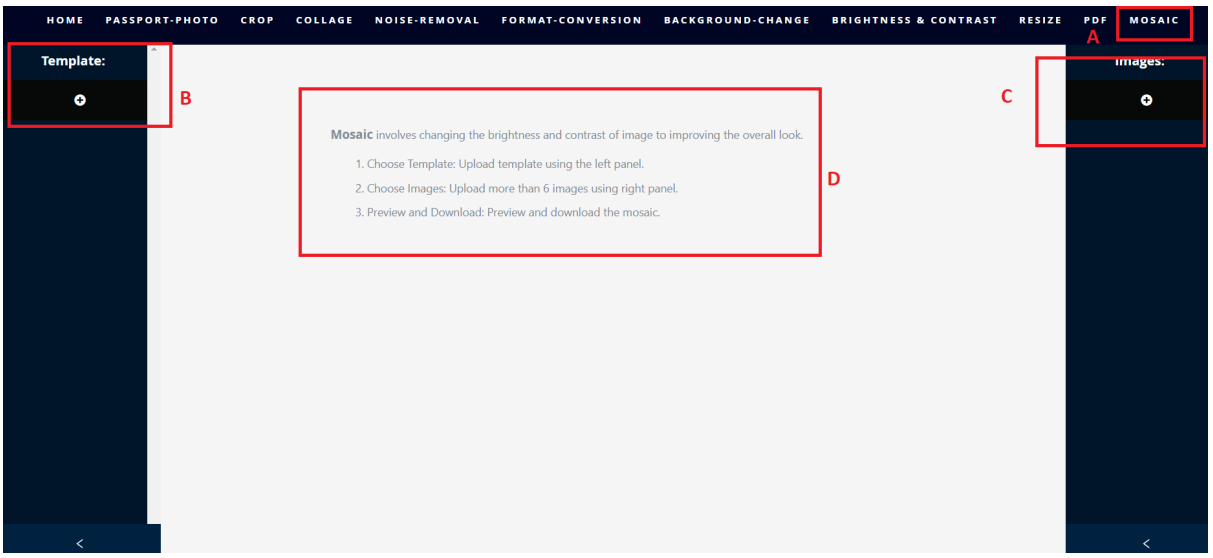



As shown in the image below, the final PDF is displayed in Region (D) once Create PDF © is clicked, and users can download the new pdf by clicking on Download button in Region (D).

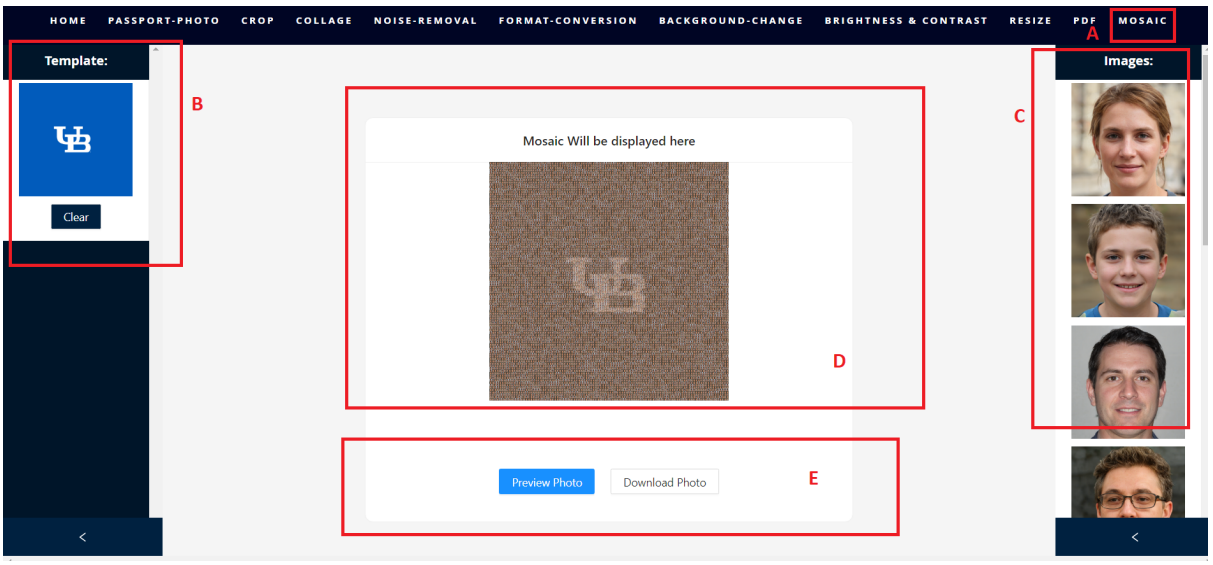



**Mosaic Creator:**

Next feature is Mosaic Creator when clicked by the user can create Mosaic out of uploaded images in Region (C) with the template uploaded as image in Region (B) and Region (D) has description on how the feature works and an area to display Mosaic.




As shown in the image below, the final Mosaic is displayed in Region (D) once Preview Photo (E) is clicked, and users can download the mosaic by clicking on Download button in Region (E).




## 8.Contribution Table:

| Team Members | Contributions |
|---|---|
| Ruthika Juluri | • Research on different available libraries that can be used to implement features with good performance.<br>- Background Change<br>- Brightness & Contrast<br>- Resize<br>- PDF maker<br>- Mosaic |

| | |
|---|---|
| | • React application using proper package.json file for respective feature libraries implementations.<br><br>• Initial Setup and Basic components creation for home page.<br>- Passport photo creation<br>- Crop<br>- Collage<br>- Noise removal<br>- Format conversion<br>- Background Change<br>- Brightness & Contrast<br>- Resize<br>- PDF maker<br>- Mosaic<br><br>• **Background Change :**<br>- Sidebar to display the color picker to which background should be changed to, description of how to use the feature, drag and drop area to display the image, preview image and respective buttons to preview, clear and download.<br>- API Integration and Handling the response image file after successful axios call and downloading the image.<br>• **Brightness & Contrast :**<br>- Sidebar to toggle brightness and contrast values, description of how to use the feature, drag and drop area to display the image, preview image and respective buttons to preview, clear and download.<br>- Handling the response image file after a successful call and downloading the image.<br>• **Resize :**<br>- Sidebar to input height and width values, description of how to use the feature, drag and drop area to display the image, preview image and respective buttons to preview, clear and download.<br>- API Integration and Handling the response image file after a |

| | |
|---|---|
| | successful axios call and downloading the image.<br>● **PDF :**<br>- Sidebar to input different images, description of how to use the feature, functional create PDF button when clicked will preview PDF with all images merged and respective buttons to download, print and perform other pdf operations.<br>- API Integration and Handling the response pdf file after a successful axios call and downloading the pdf.<br>● **Mosaic :**<br>- Left Sidebar to upload mosaic template, Right sidebar to upload multiple images, and button to create a mosaic when all required inputs are uploaded.<br>- API integration and Handling the response image file after successful call and downloading mosaic.<br>● **Back-End:**<br>- Updated code to send output file as imageUrl parameter to frontend.<br><br>● Added loader in all pages.<br>● Frontend contribution percentage - 50% |

| | |
|---|---|
| Khyathi Kunuthur | ● **Research** on available libraries that can be used to implement features with good performance<br>- Passport photo creation<br>- Crop<br>- Collage<br>- Noise removal<br>- Format conversion<br><br>● **Wireframes** for all the pages available.<br>- Passport photo creation<br>- Crop<br>- Collage<br>- Noise removal<br>- Format conversion<br>- Background Change<br>- Brightness & Contrast<br>- Resize<br>- PDF maker<br>- Mosaic<br><br>● Researched on libraries for developing the best possible UI, like react-reveal, antd and how to use them.<br><br>● **Home Page :**<br>- Home Page to display all the available features, its related descriptions and statistics.<br>- Available animations that suit the available cards in the home page and Back to top button.<br>● **Passport Photo Creation :**<br>- Sidebar to display the country dropdown box, checkbox for applying high end features, drag and drop area to display the image with crop and preview image on its side with respective buttons for preview, clear and download buttons.<br>- API Integration to the backend and handling the response image file after successful axios call.<br><br>● **Crop:**<br>- Sidebar to display the aspect ratios, description of how to use the feature, drag and drop area to |

| | |
|---|---|
| | display the image with cropper and respective buttons to preview, clear and download.<br>- Handling the response image file after successful call and download image.<br><br>● **Collage :**<br>- Left Sidebar to display all the available collage templates, Right sidebar to upload images, and area to drag and drop images on selected template.<br>- Enable drag and drop of uploaded images onto the template.<br>- Handling the response image file after successful call and downloading collage.<br>- Used CSS for creation of templates.<br>● **Noise removal :**<br>- Display description of how the feature works in order of execution, Drag and drop card area to show the uploaded image and its preview after preview button is clicked.<br>- Enable download of the final image after the user is satisfied with the preview.<br>- API Integration to the backend noise removal api and handling the response image file after successful axios call<br><br>● **Format Conversion :**<br>- Sidebar to display the available formats to be converted to, description of how to use the feature, drag and drop area to display the image and respective buttons to preview, clear and download.<br>- API Integration and Handling the response image file after successful axios call and downloading the image with the selected format.<br><br>● Added toasters in all pages to notify the user.<br>● Added info button in all pages. |

| | |
|---|---|
| | ● Frontend contribution percentage - 50% |
| Manikanta Kalyan Gokavarapu | ● **Prior Research:**<br>- I explored various photo editing websites online to gain insights into their features and incorporate exclusive functionalities into our product. This approach ensured that our product stands out by offering capabilities not found on other platforms.<br>- To implement needed features explored different image processing libraries available in python and C++ like open CV, pixelLib, dlib etc and also researched about algorithms like K-D tree and learned about different deep learning ML models like pascalvoc deep learning model for faster image processing and also to get needed results in the functionalities.<br><br>● **Feature development:**<br>Explored, Developed and Implemented 4 features from backend those are<br>**- Glasses Detection**<br>● For glasses detection we extract the nose using dlib's 64 landmark facial points and we blur the image and then find the nose bridge using edge detection and output 1 if a bridge is found or else 0 if it is not found.<br>**- Background Change**<br>● Used pixel lib and alter background libraries in combination with deep learning "load pascalvoc model" to change the background of the image.<br>- **Mosaic**<br>● First we load the template image and we do the index slicing for image using the step for rows and columns to divide the resolution.<br>● Then we preprocess the tile images and we get a mean RGB value from each tile image<br>● Then by using the KD tree algorithm we loop through each pixel in the low resolution image and extract an image that has the closest mean colour. |

| | |
|---|---|
| | ● And we replace this closest mean colour tile image in the low resolution template image pixel.<br>- Photo Collage (Backend)<br>● Used PIL and Canvas libraries to develop the photo collage feature from backend<br>● Users can input no.of images based on the chosen template and get a collage out of it directly from the backend.<br>● Developed nearly 10 templates for different images sizes chosen for photo collage feature from backend<br><br>As the photo collage needed to be interactive for the users we went ahead with the UI team's approach in the final product.<br><br>● **Testing:**<br>- Performed unit testing while building the features.<br>- After features are integrated into API and UI, Performed Functional and API testing on all the features and reported the bugs to fix.<br>● **Misc:**<br>Documentation, Poster, PPT's etc.<br><br>Overall Contribution to the project - 20%<br>Feature Development - 50%<br>Unit and Functional Testing - 50% |
| Rakesh Kumar Gavara | ● **Initial work**<br>- Initially I did an elaborate research on Django application, MVC architecture and how to efficiently implement MVC design in the project.<br>-I have created basic backend flow(Model View controller) which includes creating basic Django applications consisting of Views, Urls, Controllers and a database connection. On top of this I tried to create the rest of the API flow.<br><br>● **API development**<br>- Created APIs for<br>1. Passport photo creation |

| | |
|---|---|
| | 2. Background Change<br>3. Mosaic<br>4. Format change<br>5. PDF maker<br>6. Noise removal<br>7. Format conversion<br>8. Resize<br>- I have worked on wiring urls to appropriate controller logic.<br>- I have written views for the individual features.<br>- I have worked on throwing appropriate exceptions in error scenarios such as missing function name, wrong function name, empty input image and so on.<br>- Created separate flows within passport photo API (one flow includes high end features & other includes minimalistic features).<br>- I have researched which countries do not allow specs and if they do what aspects of specs they do not consider such as glass glare, tinted glasses and so on.<br><br>● **Challenges faced and actions taken**<br>- I have created a single API for all the functionalities but along the way while breaking down the Singleton API to individual APIs each contributing to a different functionality we got a lot of redundant code, I tried to use OOPs design to clean code the same.<br>- To maintain data privacy I have worked with the UI team(I have asked them to append a unique ID to the image at the UI end itself).<br>- I have encountered performance issues for the passport creation API. So I tried to identify the appropriate module with high computation time(background change feature). With this observation in mind and with the professor's suggestion I have added a low end version of passport creation which bypasses some of the time taking modules.<br>- I have written a script to implement a cron job without any dependency on postgres so that we can database components from our project(reduce carbon foot print). |

| | |
|---|---|
| | ● **Testing:**<br>- Added test cases for all the APIs.<br>- Performance testing, Stress testing<br>- Individual API testing, Manual end to end flow testing.<br>- Identified various bugs/improvements in UI and explained those bugs to UI team, shared my perspective, monitored their work for a day to make sure the outstanding issues are zero.<br><br>● **Misc:**<br>Prior and final documentation work,<br>Weekly presentation documents,<br>Share some templates for posters, assisted in setting up of backend code during the initial stage of testing in respective systems of the UI team.<br><br>● **Contribution metric:**<br>- Overall contribution percentage is 20%<br>Submodules -> API development :~100%,<br>Testing : 35 to 40 %. |
| Mallikarjun Rao Annam | ● **Prior Research:**<br>- I have explored various projects and after thorough discussion came up with a photo editing app idea. I have researched various design patterns to create an efficient application.<br>- I have done research on all the rules for passport photos requirements for various countries.<br>- I have researched various machine learning models to implement features such as Pose detector, Specs detector and so on. While exploring models I tried to find the ones with low computation time and high accuracy.<br><br>● **Feature development:**<br>I have researched , developed and Implemented below features from backend :<br>**- Passport photo creation**<br>● I have written a major part of the code for the passport creation tool. |

| | |
|---|---|
| | ● I have written the logic for pose detection.<br>**- Resizing**<br>● I have implemented this functionality using resize operation from open cv2.<br><br>**- PDF maker**<br>● I have written PDF maker code using the PIL library.<br>**- Noise removal**<br>● I have implemented this functionality using the bilateral filter of the opencv2 library.<br>**- Cron job(as part of postgres)**<br>● I have written the code for cron job to remove user photos from Uploads, Output and Mosaic Input folders. It uses pg_cron table of postgres as a dependency.<br><br>● **Deployment:**<br>-I have created docker scripts for backend applications and postgres. I have faced multiple issues while running docker file, under guidance of Amir I managed to create an error free docker file.<br>- I have done the local image build and pushed the image.<br>- I did the setup in IBM cloud and imported docker image and deployed it.<br><br>● **Misc:**<br>Prior and final documentation work,<br>Weekly PPTs documents,<br>Share some templates for posters, assisted in resolving dependencies as part of the requirements file. I have also done manual testing for all the flows to make sure features are working as expected.<br><br>● **Contribution metric:**<br>- Contribution percentage is 20%<br>Submodules -> Feature development :~50%,<br>Testing : 30 to 40 %. |

## 9.Suggested Changes - After Demo Day

### Front-end

- Sorted countries in the Passport Photo Creation page. - Done
- Added .env file in the front end code base. - Done
- Added CSE 611 logo to the application. - Done
- Add more format conversion options in Image Format Conversion page(facing issues if we convert images from tif, bmp, webp, pic, hdr to any other format). - Done
- Add custom aspect ratio option in Crop page. - Done
- Add instructions even during user interaction. - Done
- Make snipping in passport photo page dynamic - Not Done
- Make collage buttons visible in Collage page. - Not Done

### Back-end

- Added some more countries which restrict spects in passport creation(Note: After thorough research we identified that most of the countries allow specs in passport photo, the only catch is there should not be any glare in the specs) - Done
- Added checks to accept images of multiple formats - Done
- Find an alternative for running cron job independent of postgres - Done
- Remove postgres from the application - Done

## 10.Assumptions and Dependencies

1. Users must have a standard web browser such as google chrome or microsoft edge on their desktop.
2. The Application must use an efficient image processing library (Open CV) to handle photo editing tasks, such as cropping, resizing ,face detection etc.
3. Users must provide valid photo file formats, such as JPEG, PNG, JPG.
4. The Application must ensure the security of user data, such as photos and personal information, and comply with data protection regulations.
5. Users must have a proper internet connection on their device, so that the application runs smoothly.
6. Users must have basic knowledge of how to use web applications, as well as the process of uploading, editing, and downloading digital photos.

## 11.Appendices

List of similar products, with notes how they differ from ours:

1. Canva provides basic photo editing tools such as crop, resize, brightness and contrast adjustments but does not provide tools such as image format conversion, passport size photo creation and face detection.
2. Ribbet provides a range of basic photo editing tools, including crop, resize, color correction and collages but does not provide tools such as image format conversion, passport size photo creation and face detection.
3. BeFunky provides a range of basic photo editing tools, including crop, resize, color correction and collages but does not provide tools such as image format conversion, passport size photo creation and face detection .
4. Adobe Photoshop Express provides basic photo editing tools, such as cropping, resizing, collage and colour correction but does not provide tools such as image format conversion, passport size photo creation and face detection.
5. PicsArt also provides a range of photo editing tools, including crop, resize, collage and colour correction but does not provide tools such as image format conversion, passport size photo creation and face detection.

## 12.Discussion and comments about the course, meetings, and client

Throughout the course, we learnt about key project management concepts, such as project initiation, scope definition, scheduling, resource allocation, teamwork and quality control which play a vital role in project success. Along the journey we got better equipped with tech stack. Though we have divided the work among ourselves based on the project components such UI, Features and API we always took a peek into peers' work out of curiosity to learn and make sure we are going in the correct direction.

We have received good feedback from various people on presentation/demo day. Couple of them wanted to use our project in their blogs and so on. Two of our most appreciated features include Passport Photo creation and Mosaic. They were amazed by the research we did and we also got a positive view on our User Interface as well. We did receive some constructive feedback on background feature as the model sometimes is taking more than usual time for processing, also to add some more meta information to UI to make it more interactive.

Finally we are thankful to our Professor - JinJun Xiong and TA - Amir Nassereldine for being good critics on work which encouraged us to make our product more effective and when we get stuck or confused they have always lend their helping hand and guided us. IBM deployment process would be much more complicated & rough if our TA did not pitch in. So with the constructive feedback