Test Strategy Document

Five Fingers Innovative Solutions

## **Scope**

The document will be reviewed by Mr. Anil Kumar Upadhyay and Chitralekha Upadhyay founder of Five Fingers Innovative Solutions. They are going to approve this document after verification.

Testing activities:

* Testing of terminal based object identification from a scene with proper proper output in media folder is done in sprint 3.
* Testing of basic UI with navigation bar links and Image object identification page was done in sprint 4.
* Testing whether the image is saved in the system on uploading the image from UI is done in sprint 4.
* Testing for error handling and possible input while uploading is done in till R1.

## **Test Approach**

**Process of testing:**

● Finding all the possible test cases for a particular piece of application.

● Performing the actual testing (interacting with the system)

● Observe the result and check whether the expectations were met.

**Testing Levels:**

We have done unit testing and integration of the image object identification in GUI and UI.

**Roles and responsibilities of each team member:**

* Harshika Jain: Frontend and testing
* Subodh Sondkar: Backend of object identification and testing on GUI
* Sonu Guru: --
* Shivam Nayak: --

**Testing Approach and Types of testing:**

1. Functionality testing
   1. Check for all the links working properly
   2. Check for any broken link
   3. Check whether the image uploaded stored properly in the system
   4. Check the input format of the image
   5. If the input was wrong it should show an error
2. Usability testing
   1. The website should be user friendly
   2. The instructions provided should be clear to easy to understand
   3. The main menu should be provided on each page
   4. The output should have an option to download
   5. Functionalities should be easy to handle and can be used by general public
3. Interface testing
   1. Check the interaction between server are executed properly
   2. Check for proper error handling in case of any broken interaction

## **Test Environment**

Setup required for each environment:

* Operating system: Linux
* Front-end running environment: Python
* Back-end framework: django
* System and applications: Gitlab, VS code/sublime
* Browser: Google Chrome
* Network: local-host defined by the terminal

**Restoring strategy:**

We all are using Gitlab and we push on the gitlab after every update in the project. So we can keep a track of every commit and any commit can be restored from Gitlab directly incase we require restoring.

## **Testing Tools**

* Terminal: We test our code(backend) on terminal by executing the python file and checking whether we got an image with a highlighted object in our media folder.
* Browser: We directly test the UI for any broken link and error handling.

## **Use Cases**

* Training of model through terminal
* Output check
* WebApp
* Navigation bar links
* Image object identification page
* Input image from UI
* Wrong format image upload
* Image uploaded successfully
* Output testing

## **Test Cases**

* Give input for training the model
* Images of different scenes with/without the object to be identified
* Local host server through terminal
* Checking the links of the navigation bar
* Different format images
* Image with png extension
* Images with extension other than png