Lostrace is a Reverse Image Search based Tracking App on WCB lost Child Database, Based on Industrial Infrastructure & Highly Efficient Flex Search Engine i.e Data-points & Feature Encoding, Checkpoint Database for Omitting Repetitive Encoding, Automated Data Pipeline, Clustering to Reduce Search Span, Facial Recognition & Information Retrieval.

“Lostrace” is a solution devised by our Team-FindX to the problem in which we have to develop an app based on Artificial Intelligence and Automation that might be effective in tracking the missing child from the database of the WCD Ministry using Facial Recognition Software of Delhi Police.

According to Infographics, the number of missing children in the country has increased significantly over the last three years. Many of these cases were likely to have close links to human trafficking and child labor and we all know that child labor eventually hinders the economic and academic growth of our country, and this is the reason for our motivation to solve this problem.

The architecture of the project “Lostrace” is divided into three main parts that are App, Data API, and the Flex Algorithm.

1) **App:**
    a) Pages:
        i) Search Page
            (1) On the search page, first, the user has to authenticate himself/herself using the OTP Verification System integrated into the app. After that, the user can either capture the picture of the child using the rear or front camera or upload the image from the device which is already existing in the device. The camera feed of the app will be integrated with the Face Detection Model which will continuously detect the user’s face and thus, will eventually reduce the chances of unnecessary picture uploads loading on the server.
        ii) Processing Page
            (1) This page will show each step of the backend process happening in real-time during the Refreshing Database State, Image Encoding, Face Recognition, and Flex Searching Image.
        iii) Result Page -
            (1) The Result Page will contain the details of the missing child-like Photograph, Personal Details, Missing Place with Report Link for directly reporting to the police station, and Profile Link of the child in https://trackthemissingchild.gov.in/ which contains the detailed data of the child.
    b) Error Management:
        i) No Image Uploaded - If the image is not found on the server.
        ii) Person Not Found - If the person is not in the database.

2) **Data API**
    a) REST API: Flexible API integration with Flex Search Engine
        i) POST /search :
            (1) Initiate the Search Task with a form image
            (2) Returns Task-ID
        ii) GET /state?task_id=<_> :
            (1) Checks Task-ID Status
            (2) Returns Task Status, Failure/Processing/Success along with info msg & Details.
    b) Task-Queue Manager
        i) To avoid Server Overloading, Once Server's Simultaneous Request limit is crossed remaining Tasks in Queue are paused until prior tasks are complete.
    c) Quota Manager
        i) For Uniform Access & to Avoid Scripting, There is a Dedicated Search Quota of 1000 Search Requests Per Hour for Every User.
    d) Analytics Logger
        i) Analytics Logger Stores info about each GET & POST request on the Server, for inspection in case of malfunction. An important point to be noted is that it doesn't store the Query Image.

3) **Flex Algorithm**
    a) Data-Points
        i) We have defined a Data Point for each individual in the missing child database of the WCD ministry. These data points include the Missing ID and profile number of a particular individual so that we can re-generate all information about that person.
    b) Checkpoint-Database
        i) A checkpoint database will also be created to preserve all the above data points and the respective feature encodings of Dataset images from the last search.
    c) Data-Pipeline
        i) A data-pipelining feature is integrated to retrieve new data points and feature encodings and update the checkpoint database on a new search.
    d) Clustering-Filter
        i) The next step in the process is to filter the Top 10 Images (Face Encodings) based on the query image encodings using the K-Nearest Neighbor Clustering Algorithm. As data volumes are continuing to grow which eventually leads to increased computation time and cost therefore this step is very crucial and also one of the unique features of our solution because it eliminates the need for iterating a very large database for each query.
    e) Face-Recognition
        i) Next is the Face Recognition step which finds the best match for the query image from the Top - 10 filtered images using face encodings.
    f) Extract-Info
        i) Now all the information is extracted from the best match and is shown on the result page.

USP/Show-Stopper Features

1. Highly Efficient Computation time
   a. Checkpoint Database omits repetitive Encoding & Clustering reduces Search Span, saving Computation Time & Resources.
2. REST-API
   a. Flexible API integration with Web, App, Surveillance-Cameras, etc.
3. Handles-Load Easily
   a. With OTP-Auth & Pre Face-Detection along with a quota of 1000 Searches per Hour & Task-queue Manager.
4. Minimalistic UI
   a. Responsive Single Click Search-Mechanism