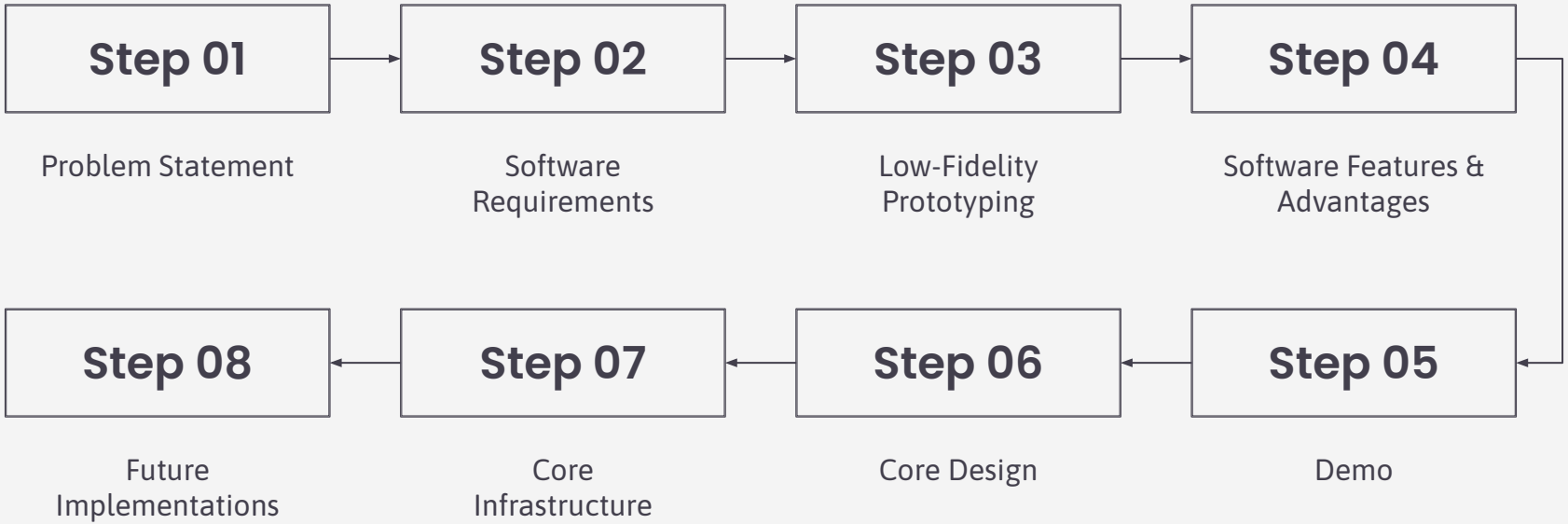
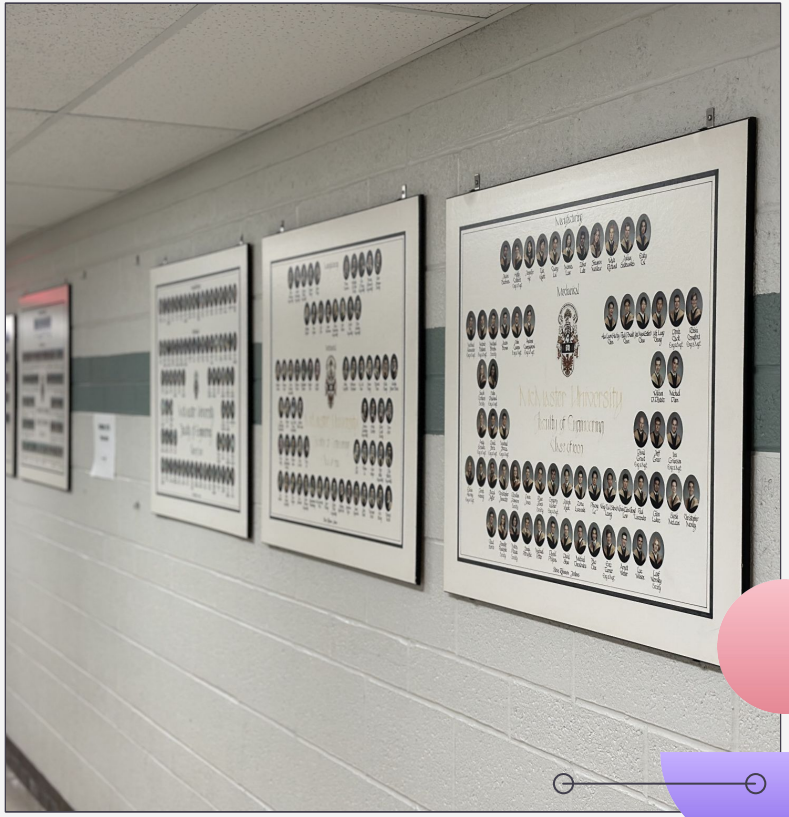
# GradSight

**Team 5**
Henushan Balachandran, Hammad Pathan,
Willie Pai, Zahin Hossain, Wajdan Faheem

# Our Roadmap


Step 01
Problem Statement
Step 02
Software Requirements
Step 03
Low-Fidelity Prototyping
Step 04
Software Features & Advantages
Step 05
Demo
Step 06
Core Design
Step 07
Core Infrastructure
Step 08
Future Implementations



# The Problem

Traditional composites are space-consuming, hard to update, and difficult to interact with

# Our Software Requirements

## Core Functionality

Users can view and search composites by name, year, program, or faculty.

## Performance & Scalability

Interface is touch-responsive, loads search results within 1–2 seconds.

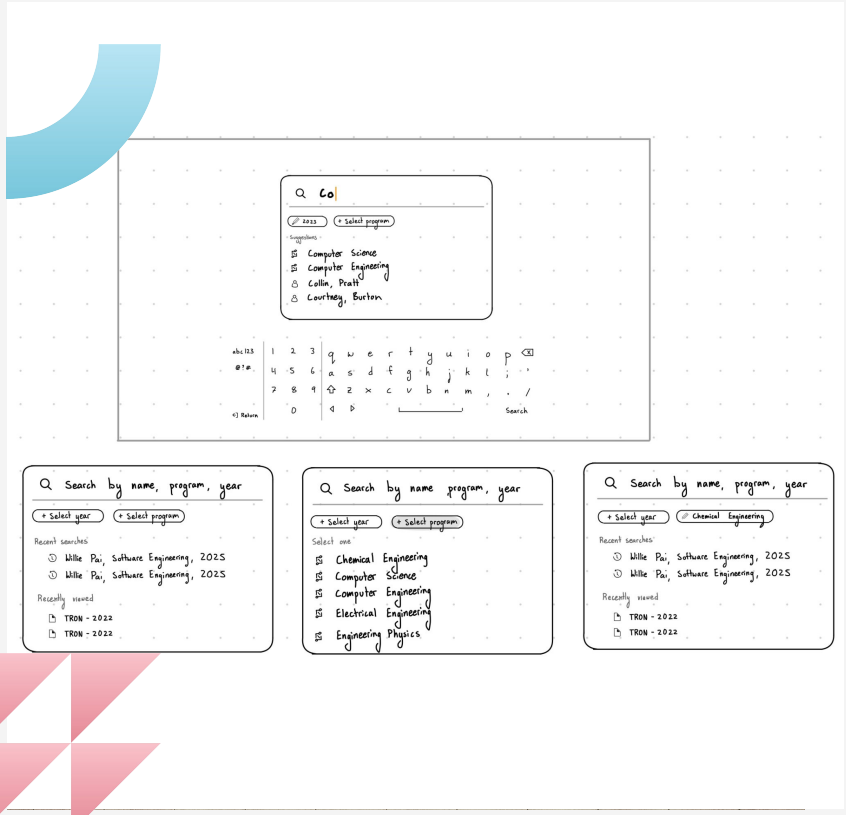
## Security & Privacy

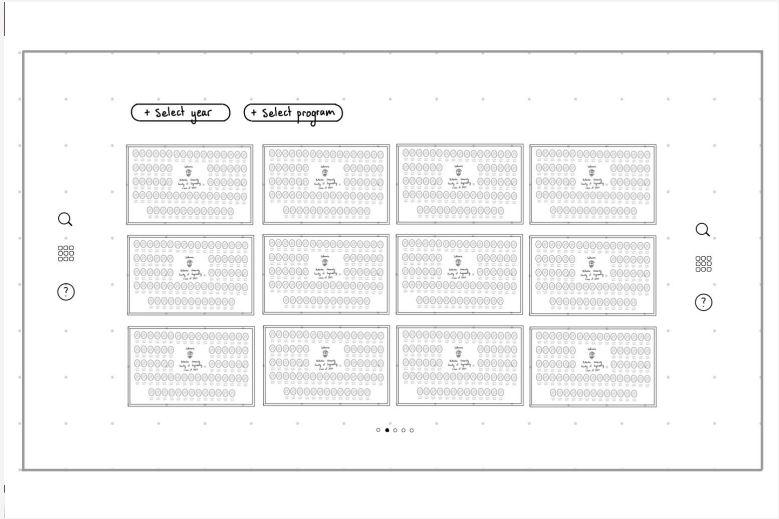
All data is encrypted, audit logs are maintained, and GDPR is followed.

## Updated SRS Compliance

Requirements are traceable (FR-1, NFR-2), and hardware constraints are documented.

# Low-Fidelity Prototyping



# What Our Service Provides...

## Minimized Space

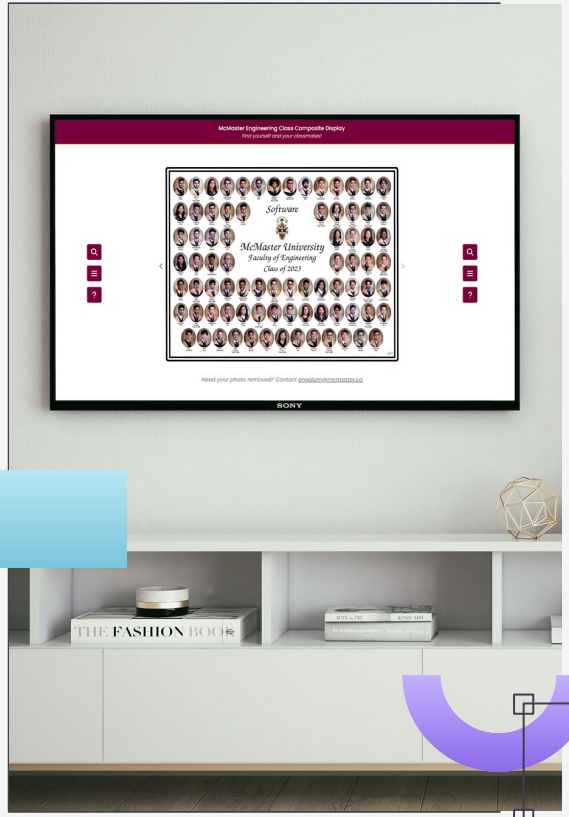
Not having to deal with physical space, you can add as many composites from any program and year without any issues

## Fast Search

Having to look through the wall and find your specific year or program is time consuming, but it is solved with a quick search

## Zoom In Capabilities

Small images can be an eyesore sometimes... By enabling zoom in option, it makes it more user friendly


McMaster Engineering Class Composite Display
Find yourself and your classmates!
Software
McMaster University
Faculty of Engineering
Class of 2023
Need your photo removed? Contact engdigimk@mcmaster.ca
SONY
THE FASHION BOOK

# DEMO

# Core Front-End Design

## Vite + React.js

User-friendly UI, responsive touch interactions

## Tailwind CSS

Styling and component utility

## Playwright

Primarily for automation of front-end testing

## Vercel

Cloud platform for deploying and hosting our frontend application

# Core Back-End Design

## Node/Express

Back-end web application framework for building API's

## AWS S3

Cloud file storage for graduate composites

## Python

OCR Integration to digitize graduate composites

## AWS RDS

Database for metadata on students and composite information

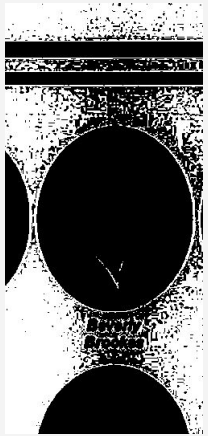
# Image Processing and OCR

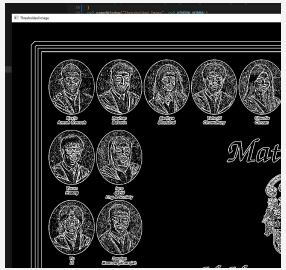
- **Manually identifying students = time consuming**

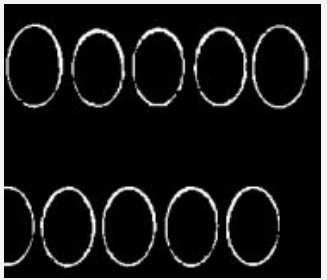
## Goal: Make algorithm that works efficiently with minimal intervention

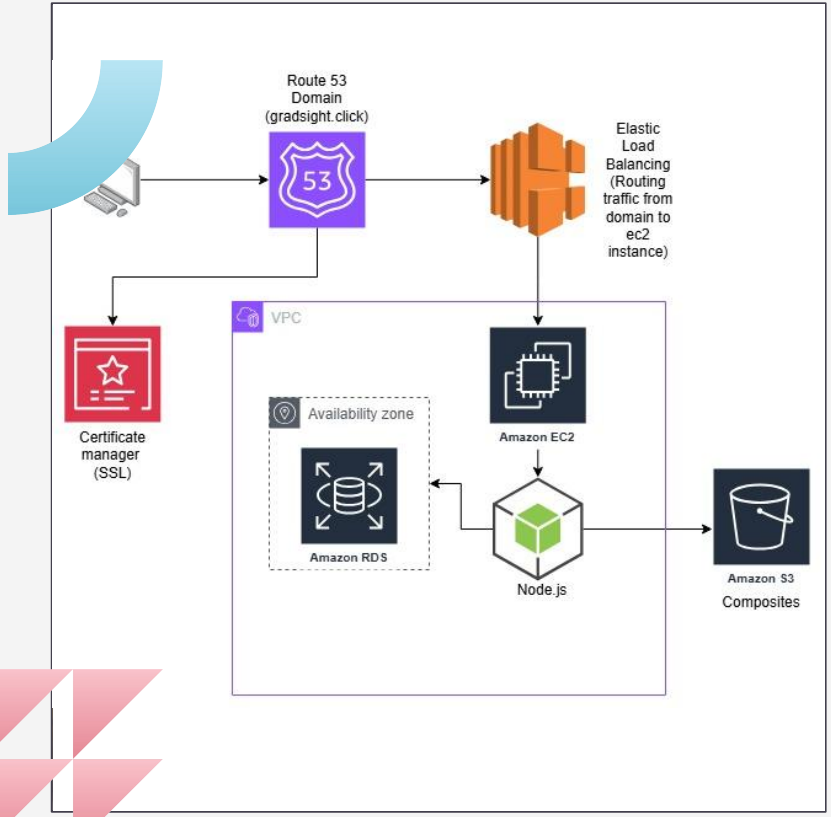
Steps (openCV module):

1. Preprocessing
2. Canny edge detection algorithm
3. Morphological ops to refine edges (dilate)
4. Identify contours
5. Filter contours
6. Extracting text regions
7. Reading with EasyOCR
8. Output student coords and names










# Infrastructure

Our application infrastructure is completely deployed on cloud using AWS & Vercel

Not sponsored ;)


McMaster Engineering Class Composite Display
Find yourself and your classmates!
Software
McMaster University
Faculty of Engineering
Class of 2023

# What Now?

# Q&A