CS651 - Web Systems
Vintage Vision

Team 8 - Haoran, Mesmer, Sravan and Navin

Overview

Web Application hosted on Google to analyze user-selected photographs from their google photos account. The analysis will include:

- Era or time when it was in fashion
- Type of fashion
- Explanation of the style
- Clickable search queries and
- Shopping tips

Technology

The app was built using the following technology:

- HTML, CSS, Javascript, React, Node.Js, Express
- Google OAuth 2.0
- Google APIs including photopicker, Vision, Gemini, Firestore/Datastore, Axios,
- Hosted on **Google App Engine Standard Environment**

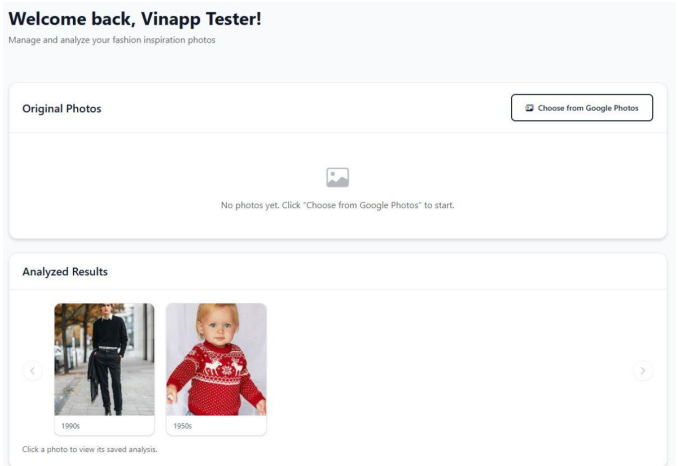
Authentication Flow

Steps:

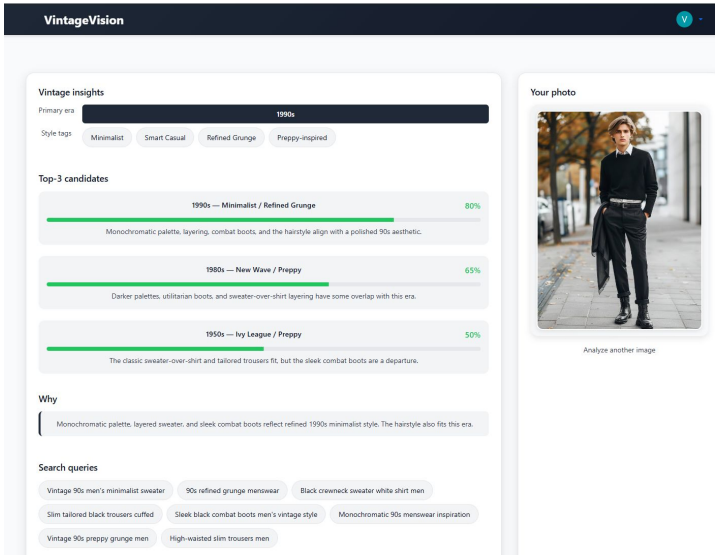
1. User signs in to their google account
2. Google OAuth end-point gets confirmation for access to user profile and photos
3. Http only cookies are stored and front end loads dashboard
4. User can view previous results or pick new photos

Dashboard

After authentication, user lands on the dashboard



Sample Analysis



Analysis Pipeline

Vision API does image detection, prominent colors, and extracts:

- Patterns, Colors, Objects

Gemini uses this information, along with image and prompt to produce:

- JSON formatted results

Firestore/Datastore

There are three JSON object collections in Firestore for this app:

1. User - authenticated users
2. Userphotos - userid, photo url, timestamp
3. Results - contains Vision and Gemini results

Caveats and Enhancements

- Caveat - Google photo urls stored in results expire after a certain amount of time leaving the dashboard with blank photo spaces for analyzed images.
- Enhancement - When user logs in, delete documents from results and userPhotos to eliminate stale urls.