DUKU AI

DUKU AI | Coding Task: Visual Change Detection MVP

Background
At Duku AI, one of the things we're building is autonomous testing that detects visual changes in web applications. For this task, you'll create a simplified version of a visual regression detection system.

The Task
Build a small web application that:

1. Takes two screenshots (before/after) as inputs
2. Compares them to detect visual differences
3. Highlights the differences visually
4. Calculates a "difference score" to indicate how much has changed

Requirements

Part 1: Backend API (e.g. Python/FastAPI or Node/Express)
Create REST endpoints for:

- POST /comparison - Accept two images and return comparison results
- GET /comparison/{id} - Retrieve a previous comparison

The comparison should:

- Calculate pixel-level differences
- Generate a diff image highlighting changes
- Return a difference percentage (0-100%)
- Store results with a unique ID

Part 2: Frontend (React)
Create a simple UI that:

- Allows uploading two images (before/after)
- Displays both images side-by-side
- Shows the diff visualization
- Displays the difference score
- Has a slider to adjust sensitivity threshold

Part 3: Bonus (if time permits)

- Add ability to ignore certain regions (draw rectangles on the image)
- Show a history of recent comparisons
- Add basic authentication with API keys

Technical Guidelines

- Use any image comparison library (e.g. pixelmatch, looks-same, or OpenCV)
- Frontend should be clean but doesn't need to be polished
- Include a README with setup instructions
- Use TypeScript if comfortable (not required)
- Include at least 2-3 test images

What We're Looking For

1. **Code Quality**: Clean, readable code with good structure
2. **Problem Solving**: How you approach the visual diff challenge
3. **Full-Stack Skills**: Ability to work across backend and frontend
4. **Pragmatism**: Making sensible trade-offs for a 3-5 hour task
5. **Documentation**: Clear README and code comments where helpful

Deliverables

- GitHub repository with your code
- README with:
  - Setup instructions
  - Any assumptions made
  - What you'd improve with more time
  - Any challenges faced

Time Expectation

This should take 3-5 hours. We're not expecting perfection - we want to see how you approach problems and structure code.

Submission

Please complete and share the Github repo link with @netproteus