Roadmap for the Project

Name : Kapil , Stream : AIML
Task : Smart Image Updater
Mentor : Mr. Akash Rikh

What is This Project ?
I’m trying to make a web tool that helps find and fix missing images in a catalog.
1.It shows a list of projects that don’t have images.
2.Fetches relevant images from the web
3.It searches for pictures of those products from the internet.
4.The tool then resizes the image to a square shape (500x500).
5.Finally, it saves the image and updates the database with the new image path.

Tech Stack :
1.Frontend – HTML,CSS,JS
2.Backend – FastAPI
3.Storage – JSON File
4.Image Preprocessing – Pillow(Python)

Working :
1. Homepage (HTML): Shows a list of products without images.
2. Search (JS → Flask): When you click "Search", it fetches mock image URLs for the product name.
3. Preview: Shows 2–3 images to choose from.
4. Select + Save: On selection, the backend:
    - Downloads the image
    - Resizes it to 500x500
    - Updates the product’s image path in the `products.json` file

5. Reload: Product disappears from the "missing image" list.

Benefits of This Approach

1.Looks better with HTML/CSS (easy to style)

2.No database needed — easy to understand and share.

3.Fully web-based and beginner-friendly.

4.Prepares you for future upgrades (e.g., adding cloud)

Project Structure :

```
smart_image_updater/
├── static/
│   └── styles.css
├── templates/
│   ├── index.html
│   └── preview.html
├── app.py
├── image_utils.py
├── fetch_images.py
├── products.json
├── requirements.txt
└── README.md
```

Functional Overview (How It Works)

1. Frontend (HTML)

    a)index.html Lists all products that don't have images.

    b)Button: "Search Image" → opens product.json for that product.

2. Backend (Flask)

    a)Loads product data from product.json

b)Uses mock image search (e.g., placeholder.com or SerpAPI if needed)

c)When user selects an image:

1.Downloads it

2.Resizes it to 500x500 using pillow.

3.Saves it locally

4.Updates the product.json in the new image path

3. Image Processing

a)image_utils.py : Makes all images square (centered and white-padded if needed)

4. Storage

a)All product info (name, code, image path) is stored in products.json

b)Saved images go inside /images.