**🧰 Tech Stack – *StyleFinder: Fashion Recommender System***

This document outlines the core technologies and tools used in the StyleFinder project.

**💻 Programming Language**

* **Python 3.8+**  
  Primary language for data processing, feature extraction, modeling, and backend integration.

**🧠 Deep Learning & Computer Vision**

* **TensorFlow / Keras**
  + Used to load and run the pre-trained **ResNet50** model for feature extraction.
  + ResNet50 is used without the top layer to extract high-level image embeddings.
* **OpenCV / PIL**
  + Image reading and preprocessing support.

**📊 Machine Learning**

* **Scikit-learn**
  + Utilized for **k-Nearest Neighbors (k-NN)** algorithm to find visually similar items based on feature vectors.

**🧪 Data Handling**

* **NumPy & Pandas**
  + Efficient data manipulation and numerical operations.
* **Pickle**
  + Used to store precomputed image embeddings (embeddings.pkl) and corresponding filenames (filenames.pkl).

**🌐 User Interface**

* **Streamlit**
  + A Python-based web app framework to create the interactive user interface.
  + Allows real-time image upload and recommendation display.

**📁 Project Structure Tools**

* **OS & Glob**
  + Directory traversal and file management.
* **TQDM**
  + Progress bars for visualization during preprocessing and model execution.

**🧪 Environment & Dependency Management**

* **pip / requirements.txt**
  + For installing and managing project dependencies.

**⚙️ Optional (for Deployment or Enhancement)**

* **Docker**
  + Containerization of the application (optional for scalable deployment).
* **Flask / FastAPI (optional)**
  + Can be used in the future to separate backend logic from the UI for API-based deployment.

**✅ Summary Table**

| **Layer** | **Technology Used** |
| --- | --- |
| Language | Python 3.8+ |
| Model | ResNet50 (Keras) |
| ML Algorithm | k-Nearest Neighbors (sklearn) |
| Data Handling | NumPy, Pandas, Pickle |
| Frontend/UI | Streamlit |
| Image Processing | OpenCV, PIL |
| File Management | OS, Glob, TQDM |
| Optional Deployment | Docker, Flask/FastAPI |