**MD SHAHIL - EXTRACT SAME TYPE OF IMAGES**

**📄 Project Requirements Document**

**Project Title: StyleFinder – Fashion Recommender System**

**1. 📘 Overview**

**StyleFinder** is a content-based fashion recommendation system that suggests visually similar clothing items based on a user-uploaded image. The system uses a deep learning model (ResNet50) for feature extraction and a k-Nearest Neighbors algorithm for similarity search, all wrapped in an interactive Streamlit UI.

**2. 🧾 Objectives**

* Enable users to upload an image of a clothing item.
* Recommend top 5 visually similar items from a dataset.
* Provide fast and accurate image processing.
* Deliver a responsive and intuitive user interface.
* Allow easy expansion for future features like filters and personalized recommendations.

**3. 🗂 Functional Requirements**

| **ID** | **Requirement Description** |
| --- | --- |
| FR1 | User can upload an image of clothing. |
| FR2 | System extracts image features using a pre-trained ResNet50 model. |
| FR3 | System searches for top-5 similar images using k-NN. |
| FR4 | Recommended images are displayed in the UI. |
| FR5 | Dataset images are preprocessed and stored in embeddings. |
| FR6 | The application runs in a web browser using Streamlit. |

**4. 🧰 Non-Functional Requirements**

| **ID** | **Requirement Description** |
| --- | --- |
| NFR1 | The system must process each uploaded image in under 5 seconds. |
| NFR2 | Recommendations must be visually similar with a high cosine similarity score. |
| NFR3 | System should be scalable for large image datasets. |
| NFR4 | UI should be clean, responsive, and user-friendly. |
| NFR5 | Embeddings must be stored persistently in a .pkl format. |

**5. 🖼 System Architecture**

**Components:**

* **Frontend**: Streamlit
* **Backend**:
  + Feature extraction (ResNet50 via Keras)
  + k-NN similarity (Scikit-learn)
  + File handling and preprocessing
* **Storage**:
  + images/: Dataset
  + uploads/: Temporary user uploads
  + embeddings.pkl & filenames.pkl: Processed data for fast lookup

**6. 🧪 Dataset Requirements**

* Images must be stored in /images/ folder.
* Acceptable formats: .jpg, .jpeg, .png
* Each image should be a single clothing item on a plain or consistent background (optional, but improves accuracy).

**7. 📦 Installation & Setup**

1. Clone the repository:
2. git clone https://github.com/SahilTwitZ/wardrobe-suggestion.git
3. cd wardrobe-suggestion
4. Install dependencies:
5. pip install -r requirements.txt
6. Place clothing images into /images/ and run preprocessing:
7. python main.py
8. Launch application:
9. streamlit run app.py

**8. 🔍 Dependencies**

* Python 3.8+
* TensorFlow / Keras
* Scikit-learn
* NumPy
* Streamlit
* OpenCV
* Pickle

**9. 🧩 Future Requirements (Optional Enhancements)**

| **ID** | **Description** |
| --- | --- |
| FTR1 | Add clothing category filters (tops, bottoms, shoes, etc.) |
| FTR2 | Integrate user-based recommendation using session history |
| FTR3 | Support mobile-friendly UI via responsive design |
| FTR4 | Offer matching outfit suggestions (e.g., shoes with shirts) |

**10. 📋 Deliverables**

* Source code (.py files)
* Pretrained model embeddings (.pkl)
* UI front-end (Streamlit App)
* This project requirements document
* Optional: Presentation/demo material