

# *DRESS DOSSIER: YOUR AI-POWERED PERSONAL STYLIST*

A Minor Project Report

Submitted in partial fulfilment of requirement of the

Degree of

**BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE &  
ENGINEERING**

BY

**PALAK HADA**

EN22CS301672

Under the Guidance of

**SANDEEP VEERWANI**



Department of Computer Science & Engineering

Faculty of Engineering

**MEDICAPS UNIVERSITY, INDORE - 453331**

APRIL-2024

## **Report Approval**

The project work **DRESS DOSSIER** is hereby approved as a creditable study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite for the Degree for which it has been submitted.

It is to be understood that by this approval the undersigned do not endorse or approve any statement made, opinion expressed, or conclusion drawn there in; but approve the "Project Report" only for the purpose for which it has been submitted.

Internal Examiner

Name:

Designation

Affiliation

External Examiner

Name:

Designation

Affiliation

## **Declaration**

I hereby declare that the project entitled “DRESS DOSSIER: YOUR AI-POWERED PERSONAL STYLIST” submitted in partial fulfilment for the award of the degree of Bachelor of Technology in ‘Computer Science & Engineering’, completed under the supervision of SANDEEP VEERWANI, Assistant Professor, Computer Science Engineering, Faculty of Engineering, Medi-Caps University Indore, is an authentic work.

Further, I declare that the content of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for the award of any degree or diploma.

**Signature and Name(s) of Student(s) with Date**

## **Certificate**

I, SANDEEP VEERWANI, certify that the project entitled “DRESS DOSSIER: YOUR AI-POWERED PERSONAL STYLIST” submitted in partial fulfilment for the award of the degree of Bachelor of Technology by PALAK HADA, is the record of work carried out by her under my guidance and that the work has not formed the basis of award of any other degree elsewhere.

SANDEEP VEERWANI

Faculty of Computer Science & Engineering

Medi-Caps University, Indore

Dr. Ratnesh Litoriya

Head of the Department

Computer Science & Engineering

Medi-Caps University, Indore

## **Acknowledgements**

I would like to express my deepest gratitude to the Honorable Chancellor, Shri R C Mittal, and my profound indebtedness to Prof. (Dr.) D. K. Patnaik, Vice Chancellor, Medi-Caps University. I thank Prof. (Dr.) Pramod S. Nair, Dean, Faculty of Engineering, for the opportunity to work on this project. I am also grateful to Dr. Ratnesh Litoriya, HOD of Computer Science & Engineering, for his encouragement and support.

This project has been made possible with their help and guidance.

PALAK HADA

B.Tech. III Year (L)

Department of Computer Science & Engineering

Faculty of Engineering

Medi-Caps University, Indore

## **Abstract**

Dress Dossier is an AI-powered wardrobe assistant that allows users to digitize, organize, and interact with their closet through a personalized virtual experience. The project aims to solve everyday fashion dilemmas by recommending suitable outfits based on occasion, weather, and style preferences. Core features include a virtual closet, outfit creation, AI-driven trend analysis, and social sharing.

Keywords: Virtual Closet, Outfit Recommendation, AI Fashion Assistant, Smart Wardrobe, Style Management

## Table of Content

		Page No.
	Report Approval	ii
	Declaration	iii
	Certificate	iv
	Acknowledgement	v
	Abstract	vi
	Table of Contents	vii
	List of figures	viii
	List of tables	ix
	Abbreviations	x
	Notations & Symbols	xi
Chapter 1	Introduction (Whichever is applicable)	
	1.1 Introduction	1
	1.2 Literature Review	
	1.3 Objectives	
	1.4 Significance	
	1.5 Research Design	
	1.6 Source of Data	
	1.7 Chapter Scheme	
Chapter 2	REQUIREMENTS SPECIFICATION	
	2.1 User Characteris	
	2.2 Functional Requirements	
	2.3 Dependencies	
	2.4 Performance Requirements	
	2.5 Hardware Requirements	
	2.6 Constraints & Assumptions	
Chapter 3	DESIGN ( Whichever is applicable)	
	3.1 Algorithm (if Applicable)	
	3.2 Function Oriented Design for procedural approach	
	3.3 <b>System Design</b> (Whichever is applicable)	
	3.3.1 Data Flow Diagrams (Level 0,Level1)	
	3.3.2 Activity Diagram	
	3.3.3 Flow Chart	
	3.3.4 Class Diagram	
	3.3.5 ER Diagram	
	3.3.6 Sequence diagram	

	<b>3.4 Database Design</b>	
	3.4.1 Logical Database Design	
	3.4.2 Physical Database Design	
Chapter 4	Implementation, Testing, and Maintenance	
	4.1 Introduction to Languages, IDE's, Tools and Technologies used for Implementation	
	4.2 Testing Techniques and Test Plans (According to project)	
	4.3 Installation Instructions	
	4.4 End User Instructions	
Chapter 5	Results and Discussions	
	5.1 User Interface Representation (of Respective Project)	
	5.2 Brief Description of Various Modules of the system	
	5.3 Snapshots of system with brief detail of each	
	5.4 Back Ends Representation (Database to be used )	
	5.5 Snapshots of Database Tables with brief description	
Chapter 6	Summary and Conclusions	
Chapter 7	Future scope	
	Appendix	
	Bibliography	
	List of Publications (If any)	
	Reprints of publications(If any)	
<b>Not</b>		



# **Chapter 1: Introduction**

## **1.1 Introduction**

In the fast-paced digital age, managing a wardrobe can be overwhelming. Dress Dossier aims to bridge the gap between technology and fashion by offering a virtual platform for outfit curation.

## **1.2 Literature Review**

Existing platforms focus on either social sharing or shopping. Few provide a unified interface to manage a personal closet with AI-driven suggestions.

## **1.3 Objectives**

To digitize the physical wardrobe

To provide outfit recommendations based on AI analysis

To integrate social sharing features

## **1.4 Significance**

The application offers time-saving solutions for daily outfit decisions and promotes sustainable fashion habits by encouraging outfit reuse.

## **1.5 Research Design**

The methodology is design-based and follows agile software development practices.

## 1.6 Source of Data

User inputs, API data for weather, and fashion trends sourced through open fashion APIs.

## 1.7 Chapter Scheme

Chapter 2 specifies system requirements. Chapter 3 discusses the design. Chapter 4 includes implementation and testing, followed by results, conclusions, and future scope.

## **Chapter 2: Requirements Specification**

### **2.1 User Characteristics**

Users include fashion-conscious individuals, working professionals, students, and anyone who wants to organize their wardrobe.

### **2.2 Functional Requirements**

Upload images of clothing items

Categorize items by type, color, fabric, etc.

Create and save outfits

Get recommendations based on weather and occasion

View and interact with fashion trends

Share looks with others and receive feedback

### **2.3 Dependencies**

Internet connectivity for data syncing and AI functionalities

Device camera and storage access

APIs for weather and trend data

## 2.4 Performance Requirements

Fast image upload and processing

Real-time outfit recommendations

Efficient handling of wardrobe data

## 2.5 Hardware Requirements

Android/iOS smartphone

Minimum 4 GB RAM and 64 GB storage recommended

## 2.6 Constraints & Assumptions

Users must manually upload clothing items

AI recommendations depend on accuracy of user-tagged data

## Chapter 3: System Design

### 3.1 Algorithm

AI model uses collaborative filtering and trend mapping from external APIs to suggest outfits.

### 3.2 Function Oriented Design

Each function like uploading, tagging, recommendation, and sharing is modular and accessed via the user interface.

### 3.3 System Design

#### 3.3.1 Data Flow Diagram (Level 0):

[User] → [Upload Clothes] → [Database] → [AI Engine] → [Recommendations]

#### 3.3.2 Activity Diagram:

Login → Upload/Tag Clothes → Explore Outfits → Get AI Suggestions → Save/Share Looks

#### 3.3.3 Flow Chart:

Start → Login → Choose Action (Upload/View/Recommend) → Output → End

#### 3.3.4 Class Diagram:

Classes: User, ClothingItem, Outfit, AIEngine, TrendFetcher

#### 3.3.5 ER Diagram:

Entities: User, Item, Outfit; Relations: owns, includes, shares

### 3.3.6 Sequence Diagram:

User → App: Upload → System: Save & Tag → AI: Suggest Outfit → User: View

## 3.4 Database Design

### 3.4.1 Logical Design:

Tables: Users, Items, Outfits, Feedback

### 3.4.2 Physical Design:

MySQL database schema with indexed keys for efficient querying

---

## Chapter 4: Implementation, Testing, and Maintenance

### 4.1 Tools and Technologies Used

Frontend: Flutter (Dart)

Backend: Firebase

AI: Python with Scikit-learn

APIs: OpenWeather, RapidAPI for fashion trends

## 4.2 Testing Techniques

Unit Testing for individual modules

Integration Testing for UI with backend

User Acceptance Testing (UAT)

## 4.3 Installation Instructions

Install via Play Store (demo version)

Enable permissions for storage and camera

## 4.4 End User Instructions



Sign up and log in

Upload images of clothes

Tag clothes appropriately

Explore outfits and suggestions

## **Chapter 5: Results and Discussions**

### **5.1 User Interface Representation**

Modern and intuitive UI using Flutter widgets

### **5.2 Modules**

Virtual Closet

Outfit Generator

Trend Analyzer

Social Sharing Hub

### **5.3 System Snapshots**

### **5.4 Backend Representation**

Realtime database structure using Firebase JSON format

### **5.5 Database Snapshots**



## **Chapter 6: Summary and Conclusions**

Dress Dossier simplifies wardrobe management and daily styling decisions. The AI component enhances user experience with smart, tailored recommendations. Overall, it proves effective as a digital personal stylist.

## **Chapter 7: Future Scope**

Integrate with e-commerce platforms for purchase suggestions

AI-based fashion scoring for outfits

Augmented reality (AR) try-on feature

Calendar integration for event-based outfit planning

## **Bibliography**

1. DressX API documentation - <https://rapidapi.com>
2. OpenWeatherMap API - <https://openweathermap.org/api>
3. Flutter Documentation - <https://flutter.dev>
4. Firebase Documentation - <https://firebase.google.com>
5. Scikit-learn Documentation - <https://scikit-learn.org>