**GRASSHOPPER-09: AI-Powered Fashion Advisor**

**Author & Developer: Lee Akpareva, MBA, MA.**

**📌 Project Overview**

**GRASSHOPPER-09** is a cutting-edge AI-powered fashion recommendation system that crafts personalized outfit suggestions based on user input. By harnessing the power of Machine Learning (ML) and Generative AI, this platform delivers fashion insights, trend-based recommendations, and interactive AI chat support.

Users can describe their desired fashion look, and the system generates both images and text-based recommendations, all while integrating an AI-driven chat feature for style guidance.

This document delves into the business strategy and technical infrastructure behind **GRASSHOPPER-09**, offering insights into its backend functionality, ML model, and overall deployment.

A close-up of a document

AI-generated content may be incorrect.

A computer screen shot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer program

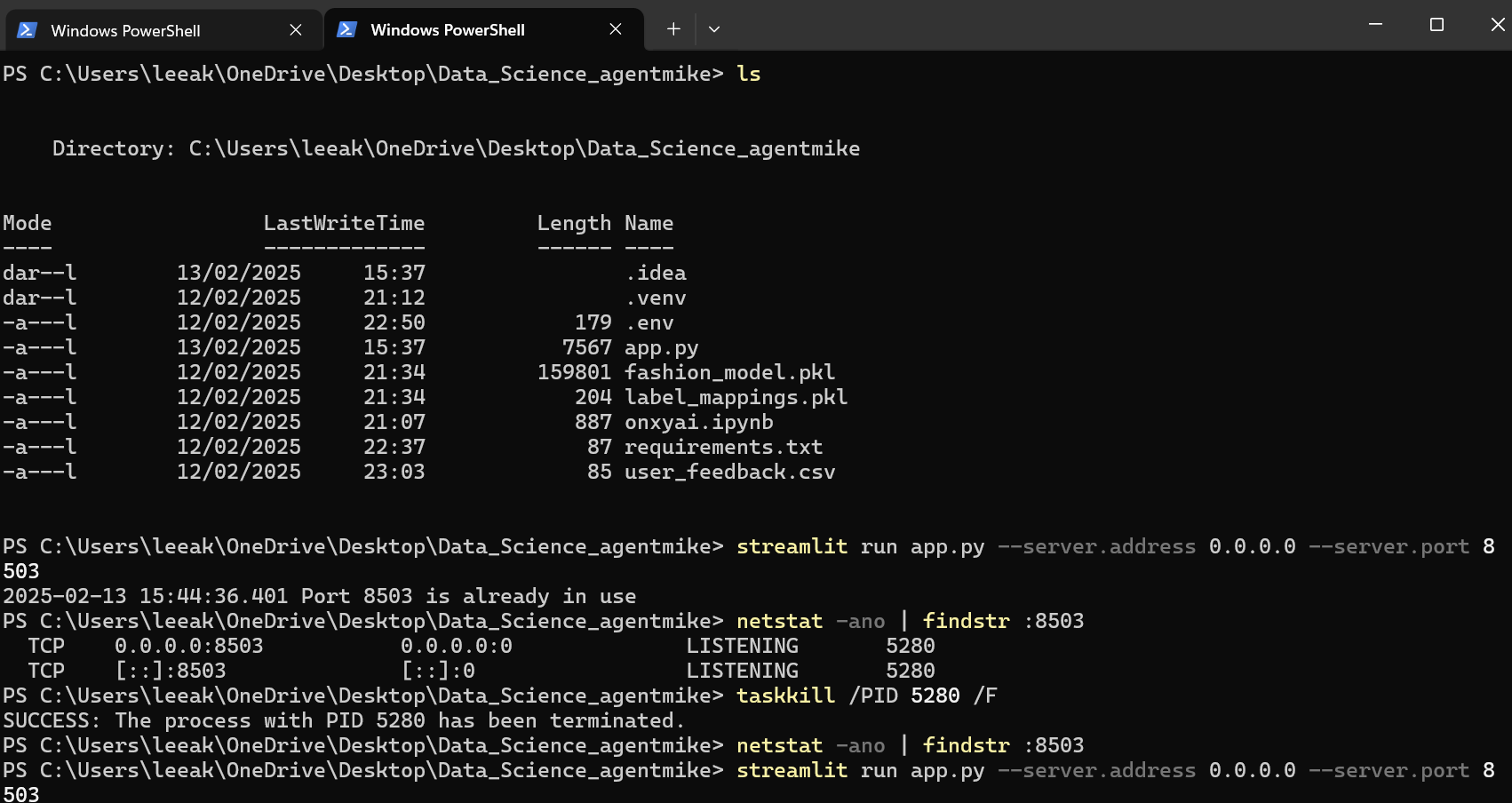
AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a website

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

**🌍 Business Infrastructure**

**📌 Target Audience**

* **Fashion Enthusiasts** looking for AI-assisted style inspiration
* **Personal Stylists & Designers** exploring digital tools for trend forecasting
* **E-commerce Platforms** interested in integrating AI-powered outfit recommendations
* **Sustainable Fashion Advocates** leveraging AI for optimized styling

**📌 Business Goals**

1. **Automate fashion outfit generation** using AI
2. **Enhance user experience** with interactive AI chat support
3. **Provide personalized styling recommendations** based on real-time user input
4. **Bridge AI and fashion technology** to create a scalable solution for designers and customers
5. **Monetize through API integrations, B2B partnerships, and SaaS offerings**

**📌 Monetization Strategy**

* **Premium API Access** for brands and stylists
* **Fashion Marketplace Integration** (Affiliate partnerships with online stores)
* **SaaS Subscription Model** for businesses looking to integrate AI recommendations
* **Ad-based revenue model** featuring sponsored fashion content

**🛠️ Technical Infrastructure**

**📌 System Architecture**

GRASSHOPPER-09 follows a **modular architecture**, ensuring scalability and efficiency:

**1️⃣ Frontend (User Interface)**

* Built with **Streamlit** for an interactive and minimalistic UI
* Fully responsive for **desktop and mobile**
* Provides user input fields for fashion idea generation and AI chat

**2️⃣ Backend (AI Processing & Data Handling)**

* **Machine Learning Model** for outfit classification and recommendation
* **Generative AI Model (DALL·E 3)** for generating fashion images
* **OpenAI GPT-4 Model** for intelligent chat interactions

**3️⃣ Database & State Management**

* Uses **Session State** in Streamlit for storing user-generated outfits
* Stores previous interactions to enhance the chat experience
* Potential future integration with **MongoDB/PostgreSQL** for user data storage

**4️⃣ Deployment & Hosting**

* **Local Testing** via http://localhost:8503
* **Network Deployment** for mobile testing with 0.0.0.0
* **Cloud Deployment Options**:
  + **Streamlit Cloud** (Free & Quick Deployment)
  + **Render** for high-performance hosting
  + **Hugging Face Spaces** for AI-focused deployment

**🤖 Machine Learning (ML) Architecture**

**📌 ML Pipeline Overview**

The backend leverages **ML models** to analyse user preferences and generate outfit recommendations. The pipeline consists of:

1️⃣ **Dataset Preparation**

* Synthetic dataset containing **fashion styles, colours, and occasions**
* Data pre-processing using **pandas**
* Feature encoding for categorical variables

2️⃣ **Model Training (Random Forest Classifier)**

* Features: Style, Colour, Occasion
* Labels: Outfit Type
* Model: **RandomForestClassifier** with n\_estimators=100
* Saves trained model as fashion\_model.pkl

3️⃣ **Fashion Image Generation (DALL·E 3 API)**

* Converts textual fashion descriptions into **AI-generated images**
* Uses OpenAI's **DALL·E 3 API** to generate visuals

4️⃣ **AI Chatbot (GPT-4)**

* Enhances user experience by remembering outfit history
* Helps users refine their selections and understand fashion trends
* Stores past interactions in st.session\_state.chat\_history

5️⃣ **Recommendation & Optimization**

* Generates a **fashion store search link** to find similar real-world outfits
* Future optimization using **Reinforcement Learning (RL)** for personalized styling

**⚙️ Code Breakdown & Backend Functionality**

**📌 Core Components**

* **ML Model Training**

from sklearn.ensemble import RandomForestClassifier

import pandas as pd

import pickle

# Sample dataset

data = {

"style": ["Casual", "Formal", "Street"],

"color": ["Black", "White", "Red"],

"occasion": ["Work", "Party", "Casual"],

"outfit": ["Blazer & Jeans", "Evening Gown", "Sneakers & Hoodie"]

}

df = pd.DataFrame(data)

# Convert categorical data to numerical labels

df["style"] = df["style"].astype('category').cat.codes

df["color"] = df["color"].astype('category').cat.codes

df["occasion"] = df["occasion"].astype('category').cat.codes

df["outfit"] = df["outfit"].astype('category').cat.codes

# Train model

model = RandomForestClassifier(n\_estimators=100)

model.fit(df.drop(columns=["outfit"]), df["outfit"])

# Save model

with open("fashion\_model.pkl", "wb") as file:

pickle.dump(model, file)

* **AI-Generated Outfit Selection**

import openai

def generate\_fashion\_image(prompt):

response = openai.images.generate(

model="dall-e-3",

prompt=prompt,

n=1,

size="1024x1024"

)

return response.data[0].url

* **AI Chatbot with Memory**

chat\_history = []

def chat\_with\_ai(user\_input):

chat\_history.append({"role": "user", "content": user\_input})

response = openai.ChatCompletion.create(

model="gpt-4",

messages=chat\_history

)

chat\_history.append(response.choices[0].message)

return response.choices[0].message["content"]

**🎯 Future Enhancements**

* **Advanced personalization** using user profile & past outfits
* **Blockchain-based digital wardrobe** for outfit ownership tracking
* **Integration with online stores** for direct outfit purchases
* **Mobile app development** for broader accessibility

🚀 **GRASSHOPPER-09 is the beginning of AI-powered fashion.** This project represents a **successful application of AI & ML in the fashion industry**, proving that technology can redefine how we experience and interact with style.