**MYNTRA FASHION ANALYSIS**

PROBLEM STATEMENT:

With the explosion of fashion products on e-commerce platforms like Myntra, there is a need for a centralized analytics tool that helps understand key trends across brands, fabrics, product categories, colours, and customer ratings. The goal is to create an interactive web-based dashboard where fashion data (in CSV format) can be uploaded and automatically analyzed for key insights.

It includes the following analysis:

* Overview
* Top brands in myntra
* Top categories preferred
* Top colours preferred
* Top fabric sold
* Top product-colour combination

**INDEX**

|  |  |  |
| --- | --- | --- |
| CHAPTER  NO | PARTICULARS | PAGE NO. |
| 01 | DESCRIPTION | 02 |
| 02 | OUTPUT | 02-05 |
| 03 | BENEFITS AND SOLUTION PLAN | 05 |
| 04 | DESIGN AND IMPLEMENTATION | 06 |
| 05 | CODE AND EXPLANATION | 06-13 |
| 06 | CLOSURE | 13 |

DESCRIPTION:

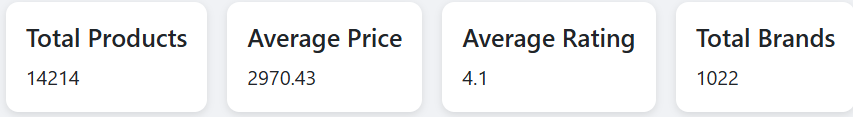
This project involves building a web application using **Flask** for backend processing and **Chart.js** for data visualization. Users can upload fashion product datasets in CSV format, and the application extracts key metrics like product count, average price and ratings, brand distribution, fabric types, and popular combinations of products and colours.

The dashboard presents the results through dynamic charts, offering users quick visual insights. The project targets fashion analysts, business stakeholders, and data science learners who want real-time feedback on product trends.

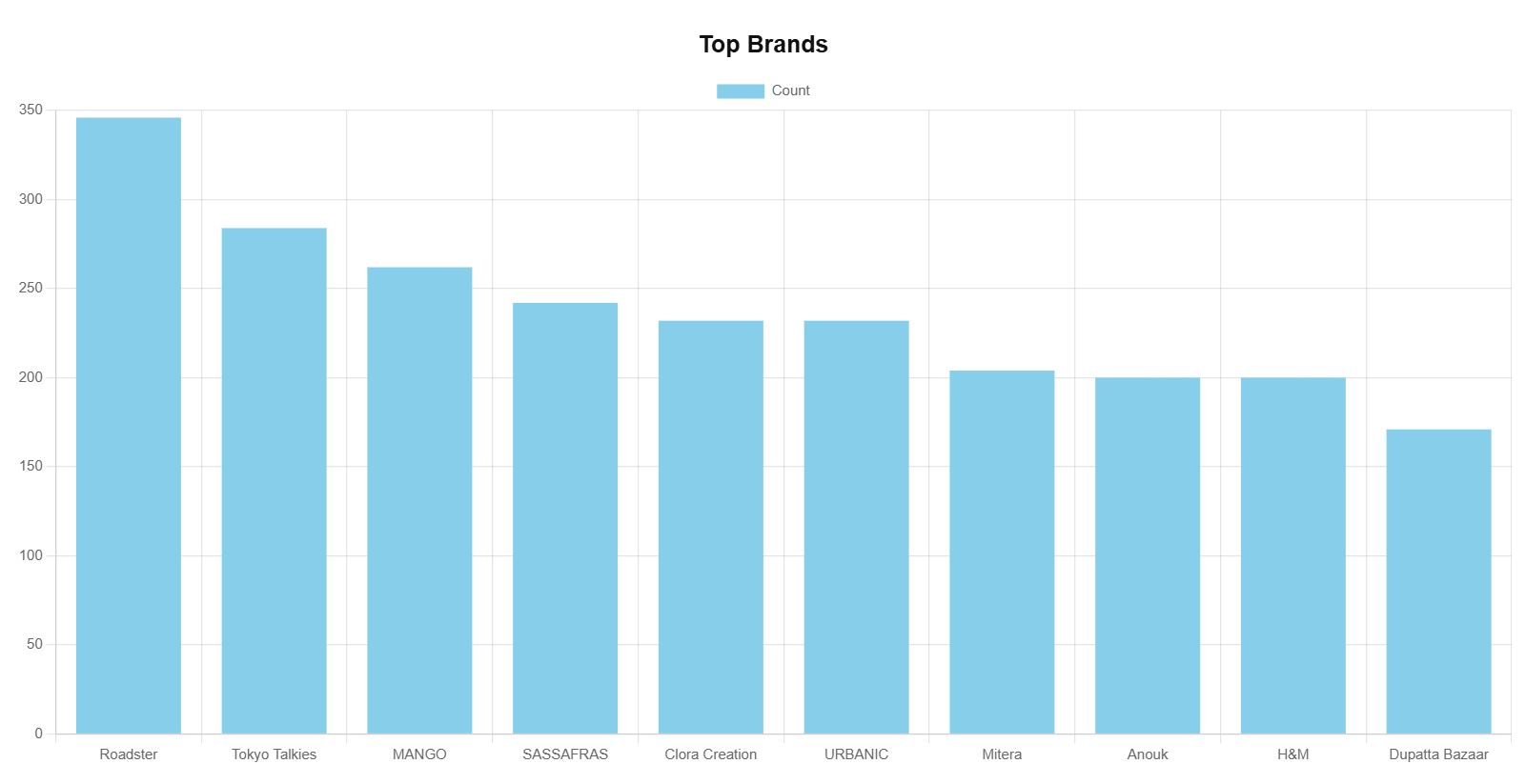
OUTPUT:

- Total products

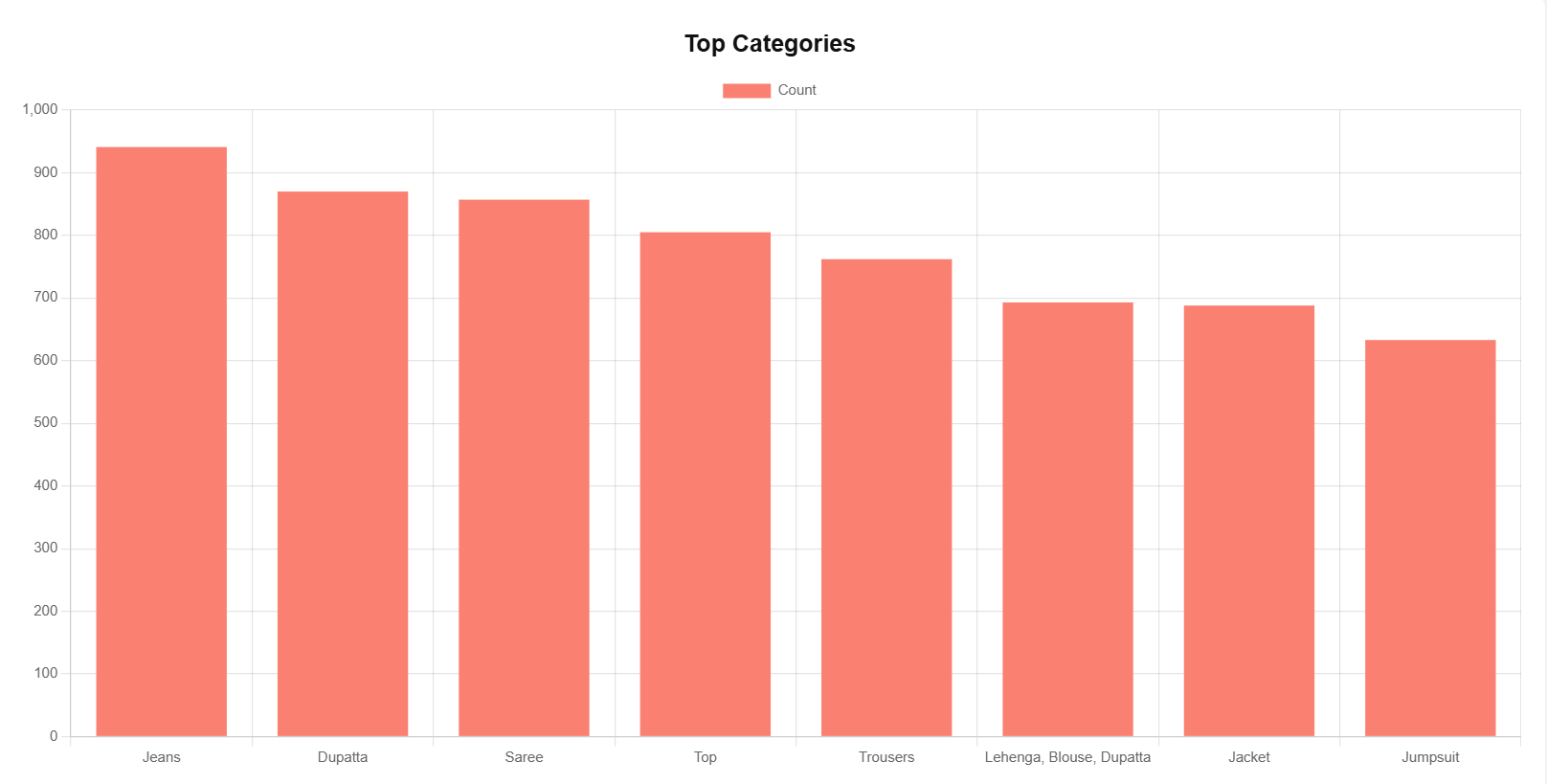
- Average price and rating



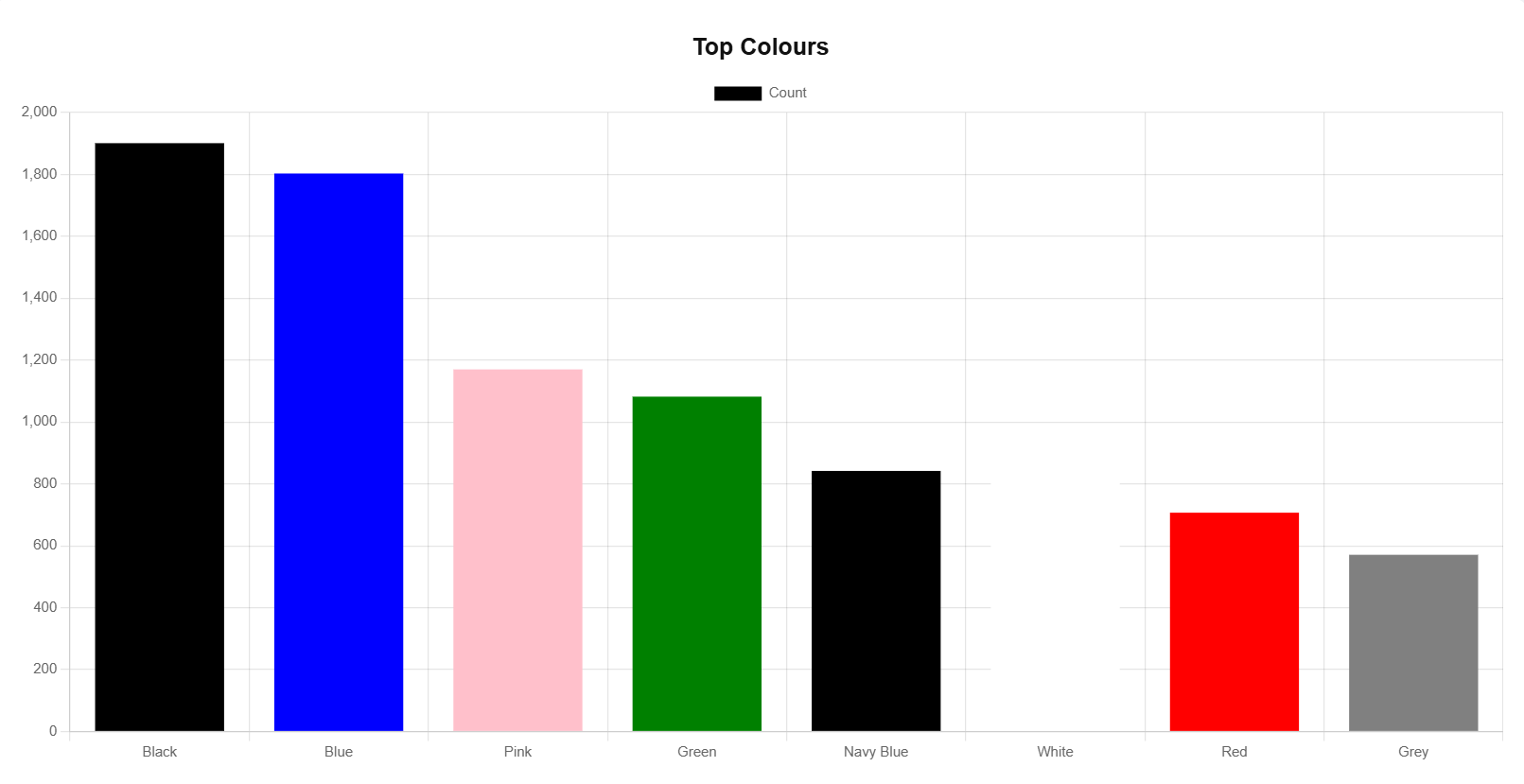
-Top 10 brands



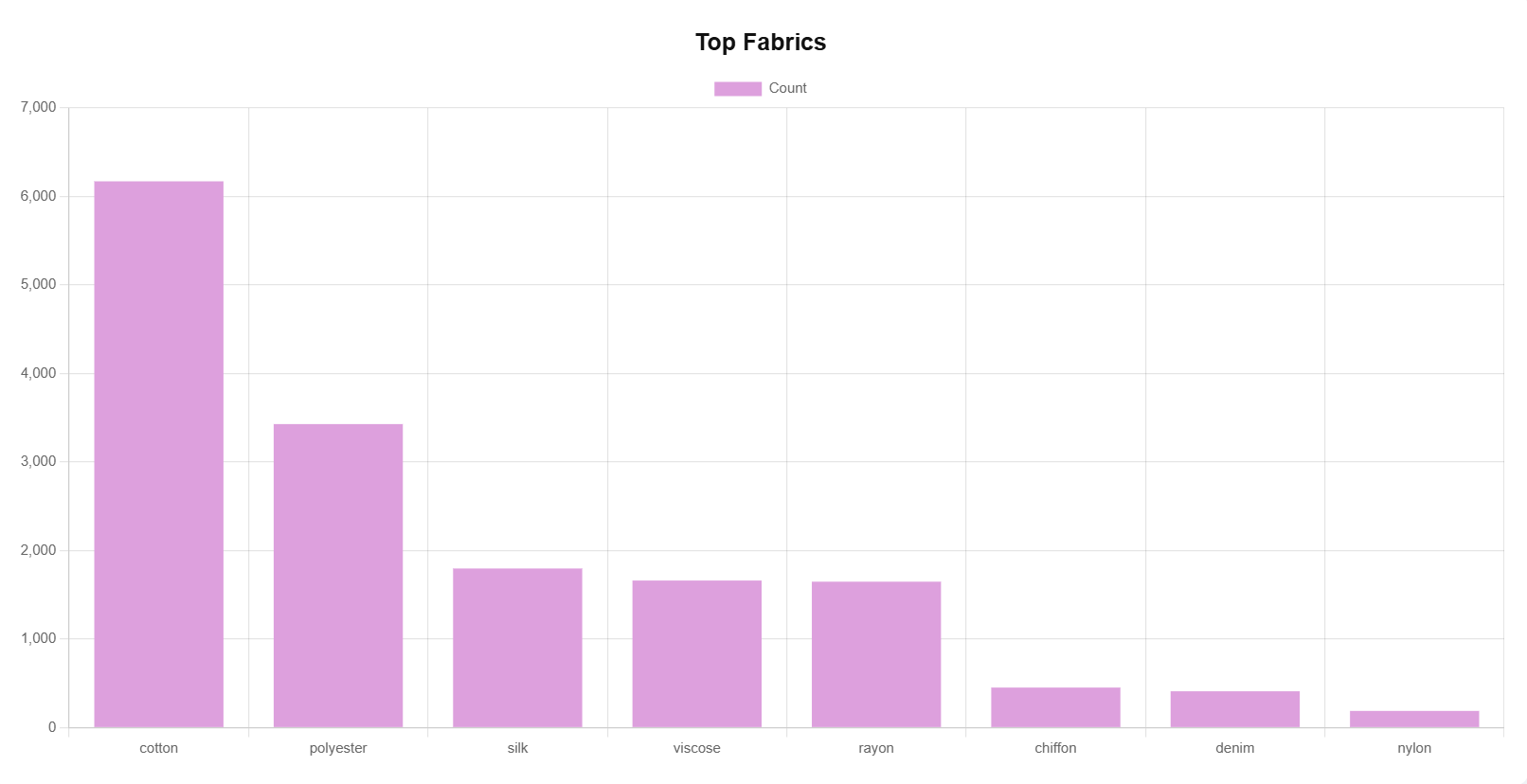
-Top 10 Categories



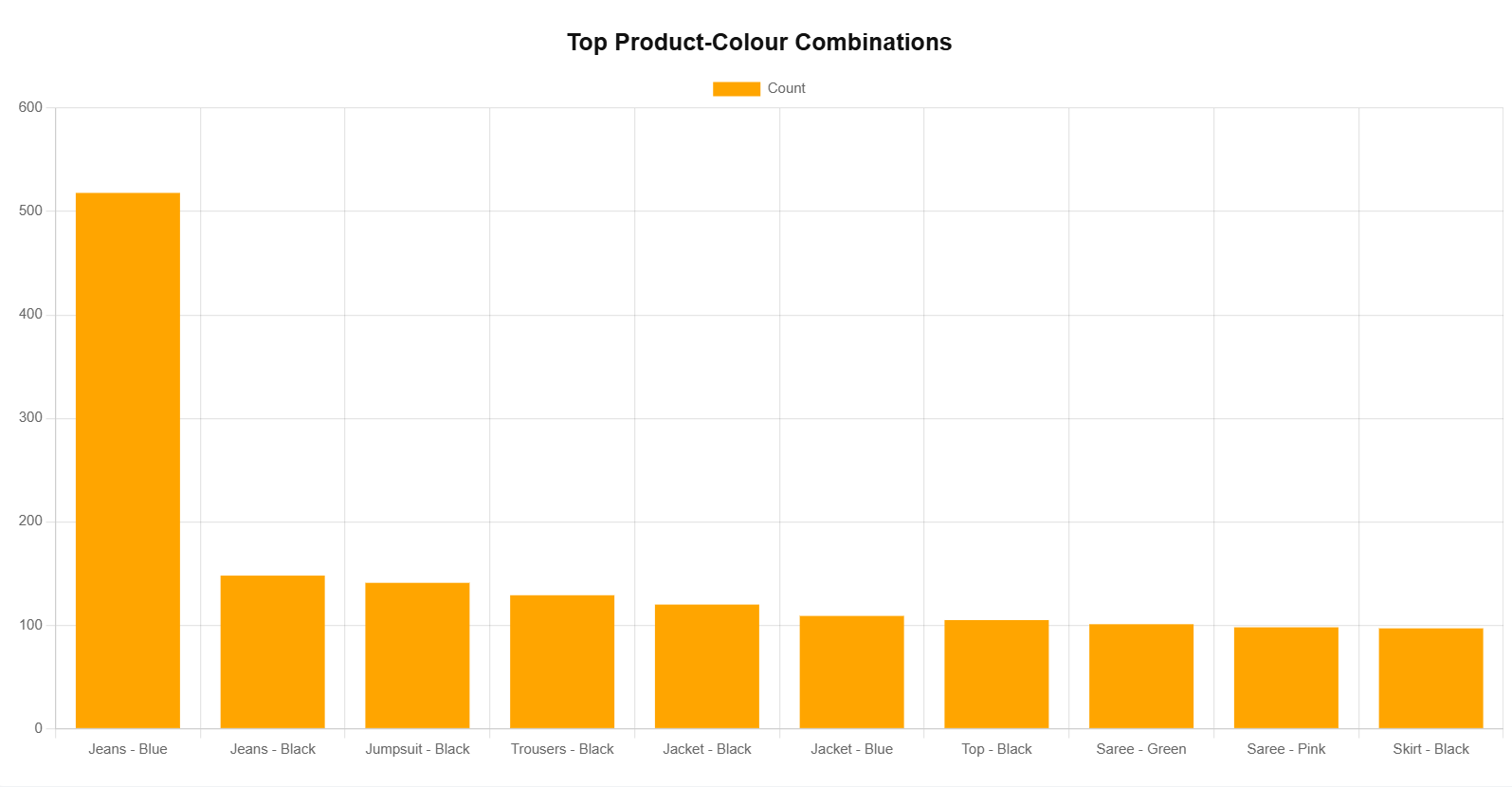
-Top 10 colours



-Top 10 Fabrics



-Top Product-Colour Combinations



BENEFITS:

- Identify trends and product popularity

- Aid inventory and branding decisions

SOLUTION PLAN:

1. CSV upload using Flask

2. Data analysis using Pandas

3. Key metrics extraction

4. Result visualization with Chart.js

5. Dashboard display

DESIGN:

Architecture:

[User Uploads CSV] -> [Flask Backend] -> [Pandas Analysis] -> [JSON Output] -> [Chart.js Dashboard]

IMPLEMENTATION:

Technologies:

- Flask (Python)

- Pandas

- HTML/CSS/JavaScript

- Chart.js

Frontend(index.html)

The HTML template uses Chart.js to render 5 bar charts. Users upload a CSV, backend processes data, and

the frontend dynamically updates metric cards and charts using JavaScript and fetch API

CODE AND EXPLANATION:

app.py

from flask import Flask, request, jsonify, render\_template, redirect, url\_for

import os

import pandas as pd

from flask\_cors import CORS

from werkzeug.utils import secure\_filename

from collections import Counter

app = Flask(\_\_name\_\_)

CORS(app)

UPLOAD\_FOLDER = 'uploads'

app.config['UPLOAD\_FOLDER'] = UPLOAD\_FOLDER

os.makedirs(UPLOAD\_FOLDER, exist\_ok=True)

@app.route('/')

def home():

    return render\_template('home.html')  # Separate landing page

@app.route('/dashboard')

def dashboard():

    return render\_template('index.html')  # Main dashboard page

@app.route('/upload', methods=['POST'])

def upload\_file():

    if 'file' not in request.files:

        return jsonify({'error': 'No file part'}), 400

    file = request.files['file']

    if file.filename == '':

        return jsonify({'error': 'No selected file'}), 400

    if file and file.filename.endswith('.csv'):

        filename = secure\_filename(file.filename)

        file\_path = os.path.join(app.config['UPLOAD\_FOLDER'], filename)

        file.save(file\_path)

        # Load dataset

        df = pd.read\_csv(file\_path)

        # Analysis 1: Basic stats

        total\_products = len(df)

        avg\_price = round(df['price'].mean(), 2) if 'price' in df else 0

        avg\_rating = round(df['avg\_rating'].mean(), 1) if 'avg\_rating' in df else 0

        total\_brands = df['brand'].nunique() if 'brand' in df else 0

        top\_brands = df['brand'].value\_counts().nlargest(10).to\_dict() if 'brand' in df else {}

        categories = df['products'].value\_counts().nlargest(8).to\_dict() if 'products' in df else {}

        # Analysis 2: Fabric types

        fabric\_keywords = ['cotton', 'rayon', 'nylon', 'polyester', 'denim', 'silk', 'linen', 'chiffon', 'viscose', 'lycra']

        fabric\_counter = Counter()

        if 'p\_attributes' in df:

            for val in df['p\_attributes'].dropna():

                text = str(val).lower()

                for fabric in fabric\_keywords:

                    if fabric in text:

                        fabric\_counter[fabric] += 1

        fabric\_analysis = dict(fabric\_counter.most\_common(8))

        # Analysis 3: Product-Colour combinations

        if 'products' in df and 'colour' in df:

            combo\_counts = df.groupby(['products', 'colour']).size().reset\_index(name='count')

            top\_combos = combo\_counts.sort\_values(by='count', ascending=False).head(10)

            top\_combinations = top\_combos.to\_dict(orient='records')

        else:

            top\_combinations = []

        # Analysis 4: Top colours

        colour\_analysis = df['colour'].value\_counts().nlargest(8).to\_dict() if 'colour' in df else {}

        insights = {

            'totalProducts': total\_products,

            'avgPrice': avg\_price,

            'avgRating': avg\_rating,

            'totalBrands': total\_brands,

            'topBrands': top\_brands,

            'categories': categories,

            'fabricAnalysis': fabric\_analysis,

            'colourAnalysis': colour\_analysis,

            'topCombinations': top\_combinations

        }

        return jsonify({'message': 'File uploaded successfully!', 'insights': insights})

    return jsonify({'error': 'Invalid file format. Only CSV allowed'}), 400

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(debug=True)

home.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Myntra Fashion Analytics</title>

    <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;600&display=swap" rel="stylesheet">

    <style>

        \* {

            margin: 0;

            padding: 0;

            box-sizing: border-box;

        }

        body {

            font-family: 'Inter', sans-serif;

            background: linear-gradient(135deg, #e6e9f0, #eef1f5);

            min-height: 100vh;

            display: flex;

            align-items: center;

            justify-content: center;

        }

        .container {

            text-align: center;

            background-color: white;

            padding: 50px 40px;

            border-radius: 15px;

            box-shadow: 0 4px 20px rgba(0,0,0,0.1);

            max-width: 600px;

            width: 90%;

        }

        h1 {

            font-size: 2.5rem;

            font-weight: 600;

            margin-bottom: 20px;

            color: #333;

        }

        p {

            font-size: 1.1rem;

            color: #666;

            margin-bottom: 40px;

        }

        a.button {

            display: inline-block;

            background-color: #6c63ff;

            color: white;

            padding: 12px 30px;

            border-radius: 8px;

            text-decoration: none;

            font-weight: 600;

            font-size: 1rem;

            transition: background-color 0.3s ease;

        }

        a.button:hover {

            background-color: #5a54e7;

        }

    </style>

</head>

<body>

    <div class="container">

        <h1>Myntra Fashion Analytics</h1>

        <p>Upload your fashion product data and explore deep insights on categories, brands, prices, fabrics, and trends.</p>

        <a href="/dashboard" class="button">Go to Dashboard</a>

    </div>

</body>

</html>

Index.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Myntra Fashion Analytics Dashboard</title>

    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

    <style>

        body {

            background-color: #f0f2f5;

            font-family: 'Segoe UI', sans-serif;

        }

        .navbar {

            background-color: #6c63ff;

        }

        .navbar-brand, .nav-link, h1 {

            color: white !important;

        }

        .container {

            margin-top: 40px;

        }

        .card-container {

            display: flex;

            flex-wrap: wrap;

            justify-content: center;

            gap: 16px;

            margin-bottom: 30px;

        }

        .card {

            border: none;

            border-radius: 10px;

            box-shadow: 0 2px 6px rgba(0, 0, 0, 0.1);

        }

        .chart-container {

            background: white;

            padding: 20px;

            border-radius: 10px;

            margin-bottom: 40px;

        }

    </style>

</head>

<body>

    <nav class="navbar navbar-expand-lg">

        <div class="container-fluid">

            <a class="navbar-brand" href="#">Myntra Fashion Dashboard</a>

        </div>

    </nav>

    <div class="container">

        <div class="mb-4 text-center">

            <input type="file" class="form-control w-50 d-inline-block" id="csvFile" accept=".csv">

            <button class="btn btn-primary ms-2" onclick="uploadCSV()">Upload CSV</button>

        </div>

        <div class="card-container">

            <div class="card p-3">

                <h5 class="card-title">Total Products</h5>

                <p class="card-text" id="totalProducts">-</p>

            </div>

            <div class="card p-3">

                <h5 class="card-title">Average Price</h5>

                <p class="card-text" id="avgPrice">-</p>

            </div>

            <div class="card p-3">

                <h5 class="card-title">Average Rating</h5>

                <p class="card-text" id="avgRating">-</p>

            </div>

            <div class="card p-3">

                <h5 class="card-title">Total Brands</h5>

                <p class="card-text" id="totalBrands">-</p>

            </div>

        </div>

        <div class="chart-container"><canvas id="topBrandsChart"></canvas></div>

        <div class="chart-container"><canvas id="categoriesChart"></canvas></div>

        <div class="chart-container"><canvas id="colourChart"></canvas></div>

        <div class="chart-container"><canvas id="fabricChart"></canvas></div>

        <div class="chart-container"><canvas id="comboChart"></canvas></div>

    </div>

    <script>

        function uploadCSV() {

            const fileInput = document.getElementById('csvFile');

            const file = fileInput.files[0];

            if (!file) return alert("Please select a CSV file");

            const formData = new FormData();

            formData.append("file", file);

            fetch('/upload', {

                method: 'POST',

                body: formData

            })

            .then(res => res.json())

            .then(data => {

                const insights = data.insights;

                document.getElementById('totalProducts').textContent = insights.totalProducts;

                document.getElementById('avgPrice').textContent = insights.avgPrice;

                document.getElementById('avgRating').textContent = insights.avgRating;

                document.getElementById('totalBrands').textContent = insights.totalBrands;

                const chartOptions = (title) => ({

                    responsive: true,

                    plugins: {

                        title: {

                            display: true,

                            text: title,

                            font: { size: 20, weight: 'bold' },

                            color: '#111'

                        }

                    }

                });

                // Sorted Top Brands

                const sortedBrands = Object.entries(insights.topBrands).sort((a, b) => b[1] - a[1]);

                const brandLabels = sortedBrands.map(item => item[0]);

                const brandData = sortedBrands.map(item => item[1]);

                new Chart(document.getElementById('topBrandsChart'), {

                    type: 'bar',

                    data: {

                        labels: brandLabels,

                        datasets: [{

                            label: 'Count',

                            data: brandData,

                            backgroundColor: 'skyblue'

                        }]

                    },

                    options: chartOptions('Top Brands')

                });

                // Sorted Categories

                const sortedCategories = Object.entries(insights.categories).sort((a, b) => b[1] - a[1]);

                const categoryLabels = sortedCategories.map(item => item[0]);

                const categoryData = sortedCategories.map(item => item[1]);

                new Chart(document.getElementById('categoriesChart'), {

                    type: 'bar',

                    data: {

                        labels: categoryLabels,

                        datasets: [{

                            label: 'Count',

                            data: categoryData,

                            backgroundColor: 'salmon'

                        }]

                    },

                    options: chartOptions('Top Categories')

                });

                // Sorted Colours

                const sortedColours = Object.entries(insights.colourAnalysis).sort((a, b) => b[1] - a[1]);

                const colourLabels = sortedColours.map(item => item[0]);

                const colourData = sortedColours.map(item => item[1]);

                const barColors = colourLabels.map(label => label.toLowerCase());

                new Chart(document.getElementById('colourChart'), {

                type: 'bar',

                data: {

                    labels: colourLabels,

                    datasets: [{

                        label: 'Count',

                        data: colourData,

                        backgroundColor: barColors  // actual bar colors match label names

                    }]

                },

                options: chartOptions('Top Colours')

            });

                // Sorted Fabrics

                const sortedFabrics = Object.entries(insights.fabricAnalysis).sort((a, b) => b[1] - a[1]);

                const fabricLabels = sortedFabrics.map(item => item[0]);

                const fabricData = sortedFabrics.map(item => item[1]);

                new Chart(document.getElementById('fabricChart'), {

                    type: 'bar',

                    data: {

                        labels: fabricLabels,

                        datasets: [{

                            label: 'Count',

                            data: fabricData,

                            backgroundColor: 'plum'

                        }]

                    },

                    options: chartOptions('Top Fabrics')

                });

                // Sorted Product-Colour Combinations

                const sortedCombos = insights.topCombinations.sort((a, b) => b.count - a.count);

                const comboLabels = sortedCombos.map(c => `${c.products} - ${c.colour}`);

                const comboValues = sortedCombos.map(c => c.count);

                new Chart(document.getElementById('comboChart'), {

                    type: 'bar',

                    data: {

                        labels: comboLabels,

                        datasets: [{

                            label: 'Count',

                            data: comboValues,

                            backgroundColor: 'orange'

                        }]

                    },

                    options: chartOptions('Top Product-Colour Combinations')

                });

            })

            .catch(err => {

                console.error("Error:", err);

                alert("Upload failed.");

            });

        }

    </script>

</body>

</html>

CLOSURE:

The Myntra Fashion Analytics Dashboard is a complete web-based data analytics tool that enables non-technical users to gain insights from product datasets. It features real-time CSV analysis, dynamic graphs, and a professional user interface.