📄 **Project Report**

Data-Driven Fashion Trend Aggregator

👤 **Author Details**

- **Name:** Sohel Rana
- **Position:** Digital Currency Investor & Technical Lead
- **Nationality:** Bangladeshi

1. Introduction

The **Data-Driven Fashion Trend Aggregator** is a FastAPI-based web application designed to collect, analyze, and recommend fashion trends to users based on their personal profiles. By integrating web scraping, AI-powered summarization, semantic search, and currency conversion, the system provides personalized fashion recommendations that adapt to user preferences, budgets, and occasions.

2. Objectives

- Aggregate fashion trends from online sources.
- Summarize trends using AI/LLM for easy user understanding.
- Allow users to create profiles with gender, style preferences, budget, currency, and occasion.
- Generate personalized product recommendations using semantic search and embeddings.
- Provide multi-currency support for global usability.

3. Project Code Structure

Code

Data_driven_fashion_trend_aggregator/

|

├── app/

| ├── __init__.py # Marks app as a Python package

| ├── main.py # FastAPI entry point (routes, app setup)

```
|   ├── database.py             # DB init, add_user, get_user, schema checks
|   ├── scraper.py              # Web scraping logic for fashion trends
|   ├── llm_engine.py           # Summarization logic using AI/LLM
|   ├── semantic_search.py      # Embedding model, currency rates, product catalog, search logic
|   |
|   ├── templates/              # Jinja2 HTML templates
|   |   ├── index.html          # Homepage (trends + summary)
|   |   ├── profile_created.html  # Confirmation after profile creation
|   |   ├── recommend.html      # Recommendations page (USD + currency conversion)
|   |   └── error.html          # Error page (user not found, etc.)
|   |
|   └── static/                 # (Optional) CSS, JS, images
|       └── style.css           # Shared stylesheet if needed
|
├── users.db                    # SQLite database (auto-created by init_db)
|
├── requirements.txt            # Python dependencies
├── README.md                   # Project documentation
```

4. System Components

Backend

- **FastAPI (**main.py**)**: Handles routes, profile creation, and recommendations.
- **Database (**database.py**)**: SQLite database with schema validation and auto-recreation.
- **Semantic Search (**semantic_search.py**)**: Embedding-based product matching with currency conversion.
- **Scraper (**scraper.py**)**: Collects fashion trend data from online sources.

- **LLM Engine (**llm_engine.py**)**: Summarizes scraped trends using AI.

Frontend

- **Templates (**templates/**)**: Jinja2 HTML templates for user interface.
- **Static Assets (**static/**)**: CSS and optional images for styling.

5. Key Features

- **Profile Creation**: Users define gender, style preference, budget, currency, and occasion.
- **Trend Aggregation**: Scrapes and summarizes fashion trends.
- **Personalized Recommendations**: Semantic search matches products to user profiles.
- **Currency Conversion**: Displays product prices in USD and user's selected currency.
- **Error Handling**: Graceful fallback when user or product data is missing.

6. Technology Stack

- **Backend Framework**: FastAPI
- **Database**: SQLite
- **AI/ML**: SentenceTransformers (MiniLM embeddings)
- **Frontend**: Jinja2 templates + HTML/CSS
- **Server**: Uvicorn

7. Future Enhancements

- Expand product catalog with real-time e-commerce integration.
- Add user authentication and profile management.
- Implement advanced recommendation algorithms (collaborative filtering, hybrid models).
- Enhance UI with modern frontend frameworks (React/Vue).
- Deploy on cloud infrastructure for scalability.

8. Conclusion

The **Data-Driven Fashion Trend Aggregator** demonstrates how AI, semantic search, and currency-aware personalization can be combined to deliver meaningful fashion insights. With its

modular structure and extensible design, the project is well-positioned for future growth and integration into larger digital ecosystems.