# CS331 Project: Retrieval Augmented Generation Assignment 1

Ahlad Pataparla — 2201017
Anushka Srivastava — 2201030
Arya Sahu — 2201033
Khushi Mandal — 2201108

January 28, 2025

## I. Feasibility study

### Technical Feasibility

- **Tools and Frameworks:**
  - **Backend:** Express for API development.
  - **Frontend:** React and Material UI for responsive web design.
  - **AI Models:** Pretrained models (e.g., CLIP for fashion item similarity, RAG model with Hugging Face libraries for information retrieval).
  - **Databases:** MySQL for storage.
- **Scalability:** Accommodate high numbers of concurrent users using scalable cloud infrastructure such as AWS and GCP.

### Schedule Feasibility

- **Timeline and Milestones:**
  - Weeks 1-2: Feasibility analysis and requirement gathering.
  - Weeks 3-5: Implementation of core retrieval and recommendation features.
  - Weeks 6-7: Testing and integration of user feedback.
  - Week 8: Final deployment.

### Quality Feasibility

- **Testing:** Continuous testing using Jest and Pytest.
- **Maintenance:** Modular codebase ensures easy updation.

### Market Feasibility

- **Demand:** Increasing trend for personalized fashion experiences in e-commerce.
- **Target Customers:** Tech-savvy individuals, particularly of the ages 18-35.

## II. Project requirements

### Customer-Provided Requirements

- A responsive e-commerce platform for fashion.
- Intelligent recommendations for outfits and similar items.

### Analyzed Requirements

- Provide natural language responses to user queries.
- Provide diverse outfit categories for personalized recommendations.

## III. Software Requirements Specification (SRS)

### a) Introduction

- **Purpose:** To build an AI-powered fashion e-commerce platform that integrates RAG for enhanced search and recommendation capabilities.
- **Scope:** Serve as a scalable and feature-rich solution catering to modern e-commerce needs.

### b) Overall Description

- **Product Features:** Outfit Recommendations, Responsive Design, Enhanced Search.
- **User Classes:** End-users (shoppers), Admins (inventory management).
- **Operating Environment:** Web browsers (desktop/mobile).

### c) System Features and Requirements

#### Functional Requirements

- **Enhanced Search (RAG)**
  - **Description:** Enables users to search for items using natural language queries, retrieving the most relevant fashion items quickly and effectively through retrieval-augmented generation techniques.
  - **Inputs:** User queries.
  - **Processing:** Retrieve relevant data and generate contextual responses.
  - **Output:** Natural language answers to user queries.

- **Matching Outfit Recommendations**
  - **Description:** Suggests complementary items to create a cohesive outfit based on the user's selected item, considering style, color, and trends.
  - **Inputs:** Selected fashion item.
  - **Processing:** AI model recommends items based on style and color.
  - **Output:** A list of matching outfits.
- **Similar Items Recommendation**
  - **Description:** Identifies and displays fashion items similar to the user's selected product, helping users explore options with comparable designs or features.
  - **Inputs:** Current fashion item.
  - **Processing:** Similarity search using CLIP or ResNet.
  - **Output:** Display similar products.

**Non-functional Requirements**

- **Performance:**
  - The system should load pages within 2 seconds.
  - Search responses should be generated in under 3 seconds.
  - The platform should handle up to 10,000 concurrent users without performance degradation.
- **Security:**
  - All user data must be encrypted both in transit and at rest.
  - Implement robust authentication and authorization mechanisms to protect user accounts and admin functionalities.
  - Regularly conduct security audits and vulnerability assessments.
  - Comply with data privacy regulations like GDPR.
- **Reliability:**
  - The system should have an uptime of 99.9%.
  - Implement robust error handling and failover mechanisms to minimize downtime.
- **Compatibility:**
  - The platform should be compatible with major web browsers (Chrome, Firefox, Safari, Edge) on both desktop and mobile devices.
  - Ensure the platform is responsive and adapts to different screen sizes and resolutions.
- **Maintainability:**

  - The codebase should be modular, well-documented, and follow coding standards to facilitate easy maintenance and updates.
  - Implement logging and monitoring to track system performance and identify issues proactively.

### d) System Attributes

- **Availability:** 24/7 service.
- **Scalability:** Built for up to 10,000 concurrent users.
- **Backup and Recovery:** Automated daily backups.

# IV. SDLC Model Choice

## Agile Model

- **Flexibility in Development:** Agile allows iterative and incremental development, enabling refinement of features such as enhanced search and recommendations as the project evolves.
- **Prioritized Customer Feedback:** Regular interaction with stakeholders helps the platform stay aligned with user expectations, especially critical for features like outfit recommendations and search.
- **Early Risk Identification:** Continuous testing and development in sprints help identify and address issues early in the project lifecycle, reducing potential delays or failures.
- **Scalability and Adaptability:** Agile easily accommodates changes in requirements or scope, such as adding new features or adjusting focus based on feasibility.
- **Enhanced Collaboration:** Promotes close collaboration among team members, ensuring developers, designers, and stakeholders are aligned throughout the process.
- **Faster Delivery:** By working in sprints, functional prototypes can be delivered earlier, allowing very quick deployment and iterative improvement.