Slide 1 – Hello & Good morning to all, welcome to our team base234’s presentation on Tansaku for designing digital enterprises. Tansaku is a Japanese word which means to search or retrieve.

Slide 2 – Let us address the problem statement here, which we’re trying to solve. If you search the web for e-commerce & shopping websites you will see in the search box is limited to text-based search only. This becomes difficult when you might have seen or come across a new or unseen pattern of designs but unaware what it is called, in such situation its hard to express in words.

Slide 3 - Take for example popular ecommerce websites – Amazon, Myntra, Flipkart you can see in the search box lacks multimodal search capabilities and only restricted to text search.

Slide 4 – Enter our solutioning product Tansaku which gives the supports for both text & visual based search capabilities leveraging state of the art generative AI technologies. This ensures more user accessibility, better user satisfaction & shopping experience from a vast catalogue of products in an efficient manner.

Slide 5 – Let us now discuss how we came up with this solutioning with our architecture diagram. We’ve designed a Flask web app having React frontend for user interaction. Once user uploads image or text query, we’ve encoded query images using the trained encoder and then find similar encoded representations from our dataset. OpenAI's CLIP (Contrastive Language-Image Pretraining) model is used for embedding text-images pairs. CLIP is suitable for text image pair-based tasks, and we’ve made use of text-image and image-image search. The model uses a ViT-B/32 Vision Transformer architecture as an image encoder and uses a masked self-attention Transformer as a text encoder. Each of these embeddings are then saved to Pinecone Vector Database which is hosted in free tier of GCP for semantic search against query. We make access request to this cloud hosted db via Pinecone Python Client package. Pinecone Vector DB consists of 512-dimension space to fit vectors. We first upsert vectors and then run query to match vectors using cosine similarity method. Thereby fetching similarity scores.

Slide 6 - This whole process is known as retrieval augmented generation or RAG.

Slide 7 – If you look at the bottom right down. We’ve also introduced a custom chatbot which uses Cohere’s Coral model to give responses based on user query such as “Tell me how to dress up for my sister’s wedding reception” or “What clothes should I pack for a summer Goa trip”. It shall come up with responses in almost no time with tips to how you can dress up for the occasion and also question you on additional details which will make it more fit for your case. It has a feature to retain chat history & thus you can continue asking until satisfied with the responses.

We’re currently also working on generating images in this chatbot. We’ve facing some computing limitations to build this as image generation models are heavier than text generation ones & thus need more processing power.

Slide 7 – Let us now head over to DEMO of the application.