<u>AI based Recommendation Approach for User Management system</u>

To improve the efficiency of the **student user management system** and enhance information retrieval for students, an AI-based recommendation system can provide significant value.

To help students more easily discover relevant academic resources, I propose implementing a **Content-Based Recommendation System powered by Sentence-BERT (SBERT)**, a transformer based model capable of understanding semantic meaning in text. Since the university offers a wide range of events, workshops, training, project archives, and research materials, SBERT can analyze these descriptions and match them to each student's interests and academic focus.

By converting both student profiles and resource descriptions into meaningful vector representations, SBERT enables accurate and personalized recommendations at scale. This approach offers an intelligent and efficient solution for helping students quickly access the most relevant academic and research information based on their interests.

<u>1) Datasets:</u>

1) <u>Student data:</u>
- Declared interests (AI, Robotics, Data Science, etc.).
- key words or short description of their project or research objectives.

2) <u>University Resource Data:</u>
- Workshops (titles, descriptions, tags
- events and seminars (abstract, topics)
- training programs
- research papers from university(abstract, author keyword)
- Project archives and past student research(summaries, problem statement, domain tags)

3) external data
- open research papers (eg, arxiv)
- public workshop or webinars
- external courses and competitions

Both student data and university resource data are considered in this system, as each contains valuable textual information that can be converted into embeddings for semantic comparison. Student inputs, such as declared interests, project descriptions, help define their academic profile. University resources including workshops, events, trainings, research papers and project archives, provide the content that the model evaluates and recommends. Additionally, the system can also include external public datasets such as open research papers, public webinars, and external courses to further expand the recommendation space and offer students broader learning opportunities beyond internal university content.

<u>2. AI model for the recommendation system</u>

I propose using a **content-based recommendation model** built on top of a pretrained **Sentence-BERT (SBERT) transformer**. SBERT converts text into semantic embeddings that capture the context and meaning of student interests as well as university resources such as events, workshops, trainings, research papers, and project archives.

For each student, the system generates an embedding based on their selected interests and project description. Similarly, each resource is transformed into an embedding using its textual description. By comparing these embeddings through a similarity measure technique such as cosine similarity, the system can rank resources and recommend the most relevant items to each student.

<u>3. Model optimization:</u>

The recommendation system can be optimized through continuous updates to the text data and through model fine-tuning. Initially, SBERT provides strong semantic embeddings without training. As more events, research papers, workshops, and project archives are added, the system becomes larger as new descriptions are converted into embeddings and included in the matching process.

In future versions, the model can learn even more effectively through fine-tuning. Using techniques such as LoRA, SBERT can be trained on pairs of university-specific texts for example, student interests paired with relevant research abstracts or events marked as similar. This teaches SBERT the language patterns, terminology, and academic structure unique to the university. Over time, this fine-tuning improves the model's ability to detect relevance, understand specific topics, and generate more accurate recommendations.

The system can also incorporate feedback signals such as which events students click on, register for, or save. These interactions can be used to re-rank recommendations or to further fine-tune the model, continuously improving the quality of results as more students use the platform.

<u>4. Performance evaluation:</u>

To evaluate the performance of the recommendation system, I would use a combination of offline metrics, user feedback, and real usage signals. First, historical or simulated student-resource pairs can be used to measure how often the system correctly identifies relevant items. Metrics such as Precision@K, Recall@K, and Mean Average Precision (MAP) can be calculated to check whether the recommended events or research papers actually match student interests.

Next, user interaction data provides strong real-world validation. By tracking which recommendations students click, save, or register for, we can measure engagement rate and compare recommendation performance over time.

Finally, direct student feedback (e.g., "Was this recommendation helpful?") can be collected to evaluate perceived relevance. Combining quantitative metrics with qualitative feedback ensures that the recommendation system remains accurate, useful and aligned with student needs.

Apart from these methods, there are several other models applicable to different recommendation scenarios, including collaborative filtering, sequence-based models, hybrid systems (such as the Netflix hybrid recommender, Spotify's hybrid approach, and two-stage recommendation pipelines), knowledge-graph embedding models, and LLM-based recommendation systems. Ultimately, selecting the right AI model depends on key factors such as the number of users, the volume and the overall scale and performance requirements of the system.