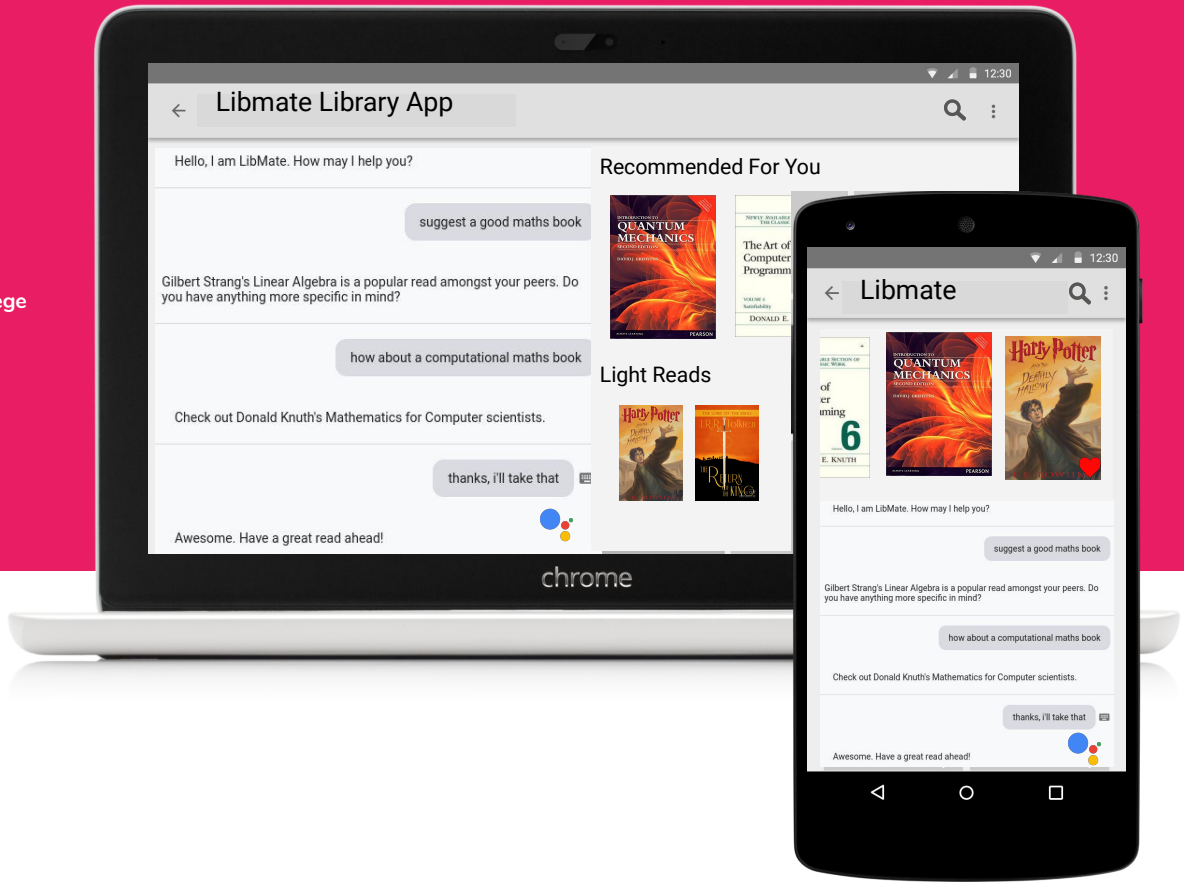
“LibMate”

Enlivening your library

Organization:	Dr. BR Ambedkar Institute of Technology
Problem statement:	Library software for Andaman College
Team name:	OccamsRazor
Team leader name:	Gaurang Tandon
College code:	1-43981653 (AICTE)

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD



Proposed Prototype

Virtual guide

- Voice assistant that guides you from aisle to aisle of the library, seamlessly through your smartphone.
- it automatically broadens and simplifies the topic of the conversation as you talk to it
- An example dialog has been shown on introductory slide.

Dependencies

- **Automatic tagger:** ***dependent*** on scraping external book services which have reasonable financial rates
- **Recommender system**: ***dependent*** data on reading patterns needs to be accumulated for few initial weeks for more accurate results

Auto Categorizer and Book Tagger

We **Scrape Wikidata, Wikipedia, and Publisher websites, as well as, use Google Books API and SparQL,** to get the data on each book i.e. its **Category Tags, Summary, Author, Publisher, Publish Date, etc**. All this is cached in an offline Database.

If we don't have complete data on a new book, we use **Machine Learning models** to generate **Vector Space Embeddings** and compare our books to all similar books based on the above metrics. Then we use these similarities to better answer searches. We perform **Hierarchical clustering** to have semantics of books retained for search and recommendations.

We will also expose API for publishers and users to **directly contribute information**. This will come with a Clean **UI for the Librarian**.

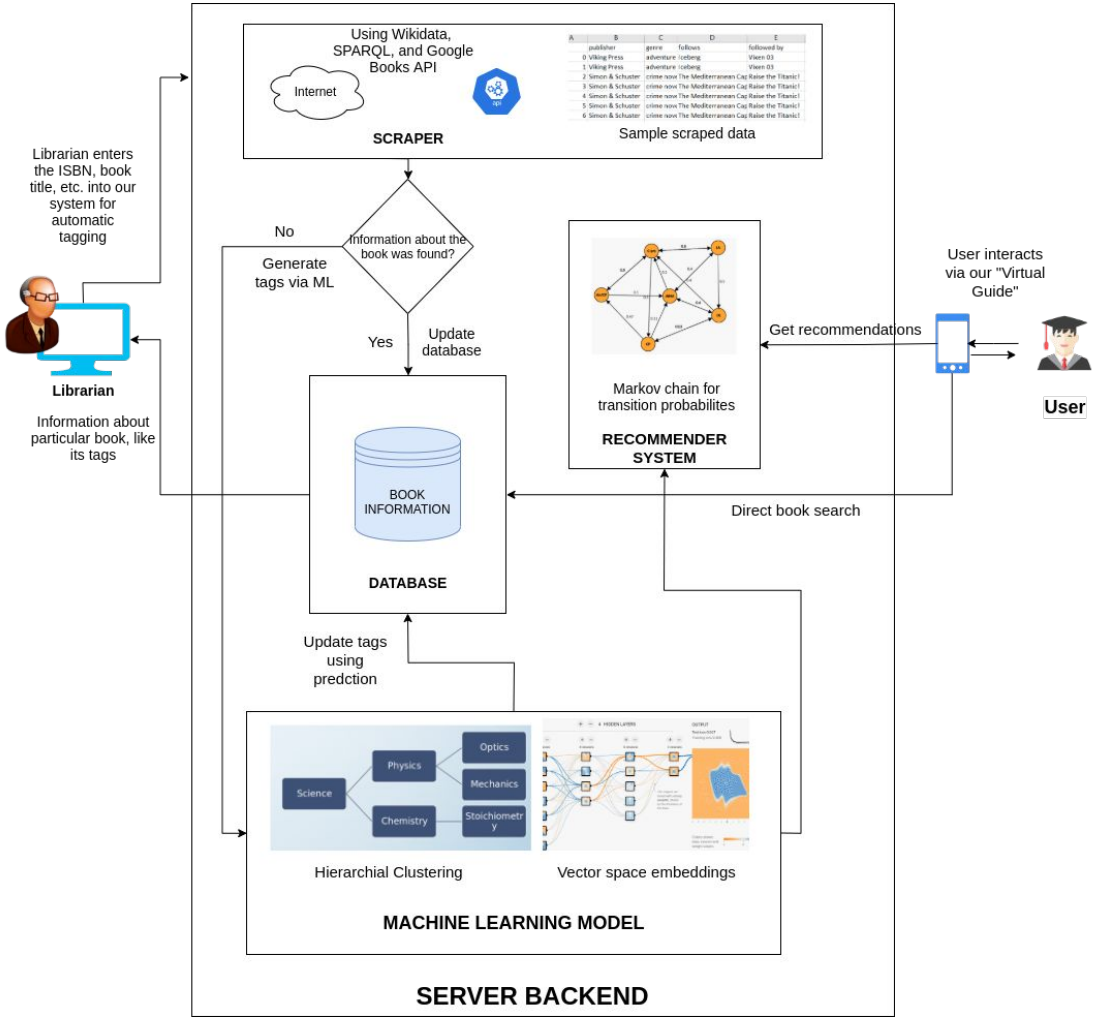
Recommended System and "BookRank"

Our "BookRank" algorithm: we believe user's reading habits can be modelled as a directed acyclic graph (**Markov Chain**), where the edges indicate the transition probability of switching.

For example, we mostly read trilogies or academic books in sequential order.

Our machine learning model dynamically learns the probability values of you reading different types of books as our Recommender application gets used, using a technique isomorphic to Value Iteration (**Q-Learning**).

This technique is similar to **Google's PageRank (Brin, Page, '98)** algorithm. But in our case, we have **Peer (learns from you and your friends) based Probability Nets.**



Tech Stack

- **Recommender system:** ML libraries
  - Google Tensorflow
    - Keras
    - Tensorflow Serving
  - Scikit-Learn
  - PyTorch
- **Mobile App**
  - React Native (UI on 1st slide)
- **APIs**
  - **Scraper**
    - Google Books API
    - Wikidata SPARQL
  - **Interface**
    - REST API for interfacing with our backend
    - Google Assistant Dialogflow
- **Backend**
  - **Server framework:** Django on Python 3
  - **Database:** Google Firebase

Use case diagram