ISP MS3-D3

By Eduard-Adiel Dinu, Elena-Bianca Dumitru, Aksel Hodne Hanisch, and Lars Huitfeldt Flem

1. Executive Summary

2. Problem to be solved

3. Background and state of the art: context and alternatives

4. Technical Project Description

    1. Solution Design
    2. Functional Architecture of the System

5. Economic Cost Analysis

6. Sustainability Analysis

7. Project Management

- Project Management Methodology and tools used
- Final IS Project Scheduling (Gantt's diagram)
- Final Task Assignment
- The IS Project Team Time sheet

8. User Manual

- Description of the Intelligent System's purpose
- Start-up/Shutdown of the system

9. Conclusions and Future Work

1. Executive Summary

Our book recommendation system presents a well-structured solution aimed at improving how users discover new books based on their preferences and reading habits. The documentation provides a detailed overview of the system's purpose, methodology, and development process while addressing its potential impact on both user experience and sustainability. The project is designed to cater to a diverse target audience, including readers seeking personalized suggestions, as well as perhaps library systems aiming to modernize their services.

At the heart of the project is a functioning prototype that leverages a database and recommendation algorithms to deliver tailored book suggestions. The recommendation methods used are collaborative filtering, content-based filtering, and natural language processing. This prototype demonstrates the potential of the system to adapt to user preferences dynamically, offering a seamless and engaging experience. While the prototype is still in development, the documentation outlines its key features, such as preference customization (rating chosen books), and interactive recommendations. The system's usability is further enhanced by a user-friendly manual, which provides practical examples and troubleshooting guidance.

The documentation also highlights the project's broader significance, including an economic analysis of its feasibility and a sustainability assessment. These sections emphasize the project's commitment to reducing resource waste, such as overstocking in libraries or unnecessary printing of promotional materials, and suggest potential cost savings for stakeholders. To strengthen this aspect, the project could integrate quantitative estimates and a comparative analysis of similar systems to highlight its competitive edge.

A clear project management strategy underpins the development process, ensuring a well-organized workflow. The use of tools like a Kanban board and Gantt chart facilitates effective task prioritization and progress tracking. This way we were able to provide a clearer picture of the timeline and dependencies. Moreover, a future-oriented approach is evident in the project's potential for scalability, such as incorporating multilingual support and expanding its genre database.

The system's target audience includes individuals and institutions that prioritize personalized experiences and technological integration in reading and education. By providing tailored recommendations and streamlining the book discovery process, the project appeals to modern readers and organizations alike. It addresses their needs for efficiency, customization, and engagement, making it a compelling proposition for implementation.

In conclusion, the book recommendation system combines technical innovation with practical utility, offering a well-rounded solution to enhance the reading experience. The prototype, supported by comprehensive documentation, demonstrates its potential to transform how users interact with literature. By focusing on personalization, sustainability, and scalability, the project not only meets current needs but also positions itself for long-term success in the rapidly evolving digital landscape.

2. Problem to be solved

The proposed book recommendation system addresses a pressing issue: the steady decline in reading habits, especially among younger generations, in an age dominated by digital distractions. Research highlights this troubling trend, with studies like the National Literacy Trust's 2024 report indicating that only 34.6% of children in the UK read for pleasure (The Guardian, 2024). This decline has far-reaching implications, limiting cognitive development, language skills, and exposure to diverse perspectives. Without intervention, reading risks becoming a niche activity rather than a cornerstone of learning and personal enrichment.

The project's primary goal is to rekindle the joy of reading by making book discovery more accessible, engaging, and relevant. By using advanced recommendation techniques: collaborative filtering (CF), content-based filtering (CBF), and natural language processing (NLP), the system offers personalized and diverse book suggestions tailored to individual preferences. User-based collaborative filtering identifies patterns among similar users, recommending books that like-minded individuals have enjoyed. Content-based filtering complements this by analyzing attributes such as genre, author, and themes to find books similar to those the user already appreciates. Meanwhile, NLP goes deeper into the textual content of book descriptions (looking for specific keywords), uncovering matches that resonate with the user's interests.

What sets this system apart is its dual focus on personalization and inclusivity. Beyond simply recommending books, the platform seeks to encourage exploration and discovery, by integrating algorithms that prioritize variety in the recommendations and introducing users to books that feature different cultures, experiences, and viewpoints. This approach not only enriches individual reading experiences but also fosters a broader appreciation for the literary world. By addressing the challenges of dwindling interest in reading, the system aspires to transform books into a compelling medium for entertainment, learning, and personal growth.

3. Background and state of the art: context and alternatives. Main competitors

3.1 State Of The Art

Recommendation systems have evolved significantly, incorporating various methodologies to provide users with personalized and relevant suggestions. Our system integrates several of these state-of-the-art methods, leveraging their strengths to address the specific challenges of book recommendation. Among the core techniques are collaborative filtering (CF), content-based filtering (CBF), and natural language processing (NLP), as well as contextual recommendations (leveraging the user's location), each contributing uniquely to the system's ability to understand user preferences and book attributes.

In this section, we will present the current methods/state-of-the-art in terms of recommendation systems:

Collaborative filtering

Most recommendation systems rely on collaborative filtering. CF recommends books based on user-to-user or item-to-item similarities (e.g., *Amazon's book recommendation engine*). However, CF faces challenges like cold starts when users or items have insufficient interactions. Also, another problem with the usual CF is the need for updating the dataset in a continuous manner to keep the new users in correlation with each other. The CF to work best requires multiple users and items to be already correlated with each other (the correlation can be measured in any way: e.g. the usual Pearson Coefficient) for a better understanding of the patterns. This is not the best method, due to its difficulty in achieving the perfect dataset, as it would require that a set big enough to be workable and scalable of users that all of them read the same books (of course in a large set) to better obtain patterns and relationships between books. Just having every user read a few books without knowing what the other set of books might offer regarding the read set is a point that must be collected.

Content-Based Filtering (CBF)

This method suggests books by analyzing their attributes (genres, authors, keywords). Systems like *Goodreads* use this by letting users rate books, followed by recommendations based on book metadata. This can be a better method than the first one presented in many situations due to the individuality of every book and user. Being a personalization task, other users' interest and preferences is not mandatory and not at all a priority. The content of the book should be the key to understanding if someone would like or not that book.

Hybrid Models

Many modern systems combine CF and CBF to mitigate their respective limitations. Netflix and Spotify are examples of hybrid recommendation engines that use a blend of algorithms to provide relevant content. Since CF mixes the preferences of different users and CBF just works with individual content, making a "voting election" between the 2 faces of the task with different weights should be achieving a better performance. This can be described as for example: a primary model with CBF that has 90% power of voting but based on various aspects regarding the user (location, age) , the second CF model can see what are the differences between that particular user and other users in the same location and age and offer more choices.

Deep Learning-Based Systems

Recent systems apply neural networks, especially recurrent neural networks (RNN) and convolutional neural networks (CNN), to model sequential patterns in user behavior and book features. These models handle complex data but may require vast computational resources. In the long term, these will be the main models that work with this task, as their disregard to handcrafted features and any logical and human behavior, will forcefully take complex patterns and tendencies in the data, that otherwise will still be found dormant with CF and CBF.

Natural Language Processing (NLP - Knowledge Based Reasoning)
NLP can be applied to analyze book reviews, summaries, or even dialogue to recommend books. This brings a more nuanced understanding of the books and can enhance the recommendation accuracy. However, this is very limited due to the need of understanding the language, so this is not a very productive method, and it will fade away very fast compared to other methods.

Social and Contextual Recommendations
Systems now consider user social connections and contextual factors (e.g., location, time, emotional state). For example, systems might suggest different books for weekend leisure compared to weekday work-related reading. This is an improvement to the usual CBF and CF, that has the potential to work really well in practice. However, due the extra information of time, emotional state, location and other relevant data, this will create irrelevant or even wrong patterns between them that will certainly occlude the real patterns.

Clustering
Most books are based on a specific structure of information and with a specific speed of developing the story. So many people say that books are similar between each other. Making user books as centroids and recommending similar books rather than relying on written specifications and preferences might be more intuitive. Regarding human points of view this is a very good method that uses a decent amount of data which does not need to be labeled (it being an Unsupervised method) but based on the uniqueness of every human being, this will fail to offer more nuanced results, unless it is paired with one or more models from the previous methods. (e.g. pairing a CBF and Clustering will reward a very good understanding of the user).

We will now present several newer methods in recommendation systems that have emerged as more advanced approaches. These methods build on traditional techniques and offer innovative ways to enhance the accuracy and personalization of recommendations.

BRAIN L (for SPANISH):
This is a new study that is combined with the normal and usual NLP. Case Based Reasoning solely relies on the logic statement that the similarity between tasks and problems implies a similarity in their solutions, that is: based on a past experience, someone can learn how to act around a similar situation. The 4 main ideas of CBR are: retrieval of information, reusing the information, revision, retain. The revision component is used to clarify the actual differences in context and to check whether a specific past solution is indeed correct or it needs alterations. In the case of alterations, the new created solution must be kept in memory for further use. As a complex model, it is bound to have better results with more data, but as big datasets are hard to obtain or hard to label, will imply a need for specialized help or just a long period of time dedicated to processing the dataset.

BOOK RS and AI POWERED SEARCH:
The literature survey on book recommendation systems and AI-powered search methodologies emphasizes the evolution of various filtering models aimed at enhancing user experience in discovering literature. It explores the complexities of personalizing book suggestions through collaborative filtering, content-based filtering, and innovative deep learning approaches, highlighting the importance of user profiling and feedback mechanisms. Additionally, the paper addresses AI's role in refining search processes, particularly through semantic search techniques and ranking functions such as BM-25, which improve the relevance of results by understanding user intent.

3.2 Main Competitors

	Goodreads	Amazon
Strengths	Large community, detailed reviews	Advanced algorithms, personalized
Weaknesses	Outdated UI, limited personalization	Commercial bias, overwhelming options
Personalization	Based on ratings and lists	Based on purchase and browsing data
Social Features	Reviews, book clubs, reading lists	Minimal, mostly product reviews
Recommendation Bias	Popular books dominate	Skewed by sales and promotions
Platform Bias	Book discovery and social engagement	E-commerce and book sales

In the competitive landscape of book recommendation systems, platforms like Goodreads and Amazon stand out as major players. These systems leverage different strengths and face unique challenges in terms of user experience, personalization, and social features. Their approaches to book discovery and recommendations have shaped the expectations of readers, but they also present opportunities for innovation and improvement.

(Ref to a study comparing book recommendation systems)

Table 1: Comparative Study of Different Algorithms for Book Recommendation Systems

YEAR	AUTHOR(S)	PURPOSE	METHOD MENTIONED
2016	Ms. Praveena Mathew, Ms. Bincy Kuriakose, Mr. Vinayak Hedge	Book Recommendation System Through Content-Based And Collaborative Filtering Method	Combined Features Of Content-Based Filtering (CBF), Collaborative Filtering (CF), and Association Rule Mining; ECLAT
2018	Yongen Liang, Shiming Wan	The Design And Implementation Of Books Recommendation System	Collaborative Filtering
2013	Dharmendra Pathak, Sandeep Matharia, C. N. S. Murthy	Nova: Hybrid Book Recommendation Engine	Combination Of Collaborative, Content And Context-Based Recommendation Algorithms
2018	Ahmed M. Omran	A Novel Recommender System For Websites	Hybrid Method
2017	Adli Ihsan Hariadi, Dade Nurjanah	Hybrid Attribute And Personality Based System For Book Recommender	Hybrid-Based Method
2014	Anand Shanker Tewari, Abhay Kumar, Asim Gopal Barman	Book Recommendation System Based On Combined Features Of Content-Based Filtering, Collaborative Filtering And Association Rule Mining	Combined Features Of Content Filtering, Collaborative Filtering And Association Rule Mining; Web Usage Mining
2009	Binge Cui, Xin Chen	An Online Book Recommendation System Based On Web Service	Content-Based
2015	Kumari Priyanka, Anand Shanker Tewari, Asim Gopal Barman	Personalized Book Recommendation System Based On Opinion Mining Technique	Content-Based and Opinion Mining Technique
2000	Raymond J. Mooney, Loriene Roy	Content-Based Book Recommending Using Content-Based & Text Categorization	Content-Based & Text Categorization
2017	Youdong Yun, Danial Hooshyar, Jaechoon Jo, Heuiseok Lim	Developing A Hybrid Collaborative Filtering Recommendation System With Opinion Mining On Purchase Review	Collaborative Filtering With Opinion Mining
2020	Jian Shen, Tianqi Zhou, Lina Chen	Collaborative Filtering-Based Recommendation System For Big Data	Collaborative Filtering

Table 1: *Comparative study between methods from Ref [5]*

4. Technical Project Description

4.1. Solution Design

4.1.1. Task Analysis

- WP 1: Backend functionality
  - Goal 1: Handling the data
    - Find accurate datasets for the project
    - Combine relevant datasets
  - Goal 2: Implement the methods
    - Make user based collaborative filtering
    - Make content based collaborative filtering
    - Make NLP recommendations using keyBERT
    - Make TD-IDF matrix for keyBERT
  - Goal 3: Combine content based, user based and NLP methods

        - Start by combining content based and user based methods
        - Combine all methods together
        - Find accurate weights for that provides a good balance between the methods

- WP 2: Frontend functionality
    - Goal 1: User Interface (UI) Design
        - Create wireframes and prototypes for the application.
        - Ensure the design is intuitive, visually appealing, and user-friendly.
    - Goal 2: User Profile Management
        - Allow users to create, edit, and delete profiles.
        - Display personalized recommendations based on user preferences.
    - Goal 3: Search and Filter Options
        - Enable users to search by title, author, genre, or keywords.
        - Provide advanced filtering (e.g., by rating, publication date, or availability).
    - Goal 4: Ratings and Feedback
        - Allow users to rate books and give feedback to refine recommendations.

4.2. Functional Architecture of the System. Technologies. User Flow

4.2.1. Technologies

The application itself was built using Flask, a flexible and lightweight web framework that is well-suited for small to medium-sized applications. Flask served as the backbone of our web application, allowing us to rapidly develop and deploy the system with minimal overhead. The simplicity of Flask made it an ideal choice for our prototype, as it enabled us to focus on building the core features without the complexities of a more heavyweight framework.

In our application, Flask handled both the frontend and backend components. The frontend consists of multiple pages that guide the user through the process of signing in, selecting books, rating them, and receiving recommendations. Flask routes handle the user navigation, displaying the appropriate pages based on user input and actions.

When a user signs in, inputs their details (name, age, location), and selects books to rate, the frontend sends these inputs to the backend via HTTP requests. Flask processes these requests and updates the backend database with the user's ratings. This data is then used by the recommendation engine to compute personalized suggestions for the user.

The backend in Flask serves as the bridge between the user input and the recommendation engine. It receives the ratings, integrates them into the recommendation model, and computes the necessary user-item matrices. Once the calculations are done, Flask sends the recommendation data back to the frontend, where the user sees the tailored list of suggestions.

4.2.2. Algorithm Explanation

The book recommendation system integrates multiple methods and technologies to provide users with personalized and diverse book suggestions. One of the key components of our system is User Based Collaborative Filtering (UCF), which is widely used in recommendation engines to provide personalized recommendations based on user behavior. We utilized cosine similarity to measure the similarity between users based on their rating patterns, and scikit-learn's StandardScaler was used to normalize the user-item matrix for better performance. This allows the system to find similar users and recommend books based on the preferences of like-minded individuals. Additionally, we made use of pandas to process the dataset, organizing the ratings and user information into a user-item matrix, while SciPy helped handle the sparse matrix efficiently, ensuring that our system can scale with large datasets.

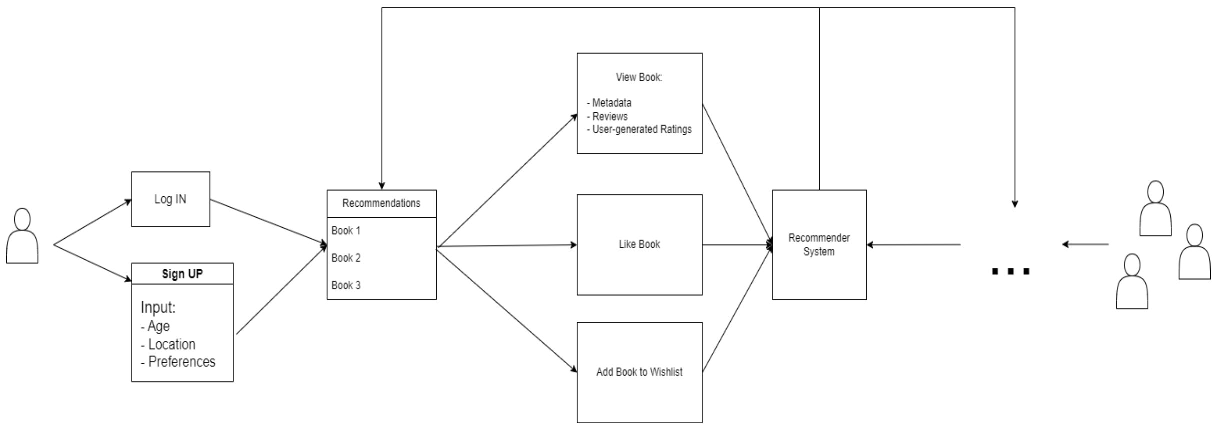
Another crucial aspect of our recommendation engine is Content-Based Filtering (CBF). This method recommends books based on the similarity of books. CBF helps personalize recommendations for each user, focusing on the content of the books they've rated highly in the past. By considering individual preferences, the system can suggest books that share similar attributes with those the user has already rated highly.

Incorporating Natural Language Processing (NLP) into our system enhances the recommendations by analyzing book descriptions.. This allows the system to understand the content of the books more deeply, providing additional personalized suggestions based on the textual content. NLP contributes to making the recommendation system smarter by taking into account not only user behavior but also the semantic meaning of the books' content. We use keyBERT on the description of the books to tag the different descriptions. Then we return the most similar books based on the description.

Using a Hybrid Approach we combine the different methods.Each method returns a certain amount of recommended ISBNs which are scored based on how good the recommendation is. Then these scores are normalized and combined using weights for each method. If a book is present in more than one method, it will appear higher on the list. The weights we have found to work well in our case, balancing between the different methodologies:

- **user_based_weight = 0.3**
- **conten_basedt_weight = 0.35**
- **nlp_weight=0.35**

4.2.3. User flow



The user flow in our system is designed to be simple and intuitive, allowing users to easily navigate through the process of signing in, rating books, and receiving personalized recommendations. Upon signing in, the user provides their name, age, and location, which is then used to tailor recommendations based on contextual factors such as age and location. This information helps refine the recommendations, making them more relevant.

After signing in, the user is directed to the second page, where they can search for and select books they want to rate. This step allows users to browse a wide variety of books and select the ones they are familiar with or interested in. Once the user has chosen the books, they move on to the next page, where they rate the selected books on a scale from 1 to 5. These ratings form the basis of the recommendations the system will generate for them.

The final page displays the personalized recommendations, which are generated by blending three different approaches: user-based collaborative filtering, content-based filtering, and NLP. These recommendations are calculated based on the user's ratings, the preferences of similar users, and the content of the books they have rated highly. The user is then presented with a combination of books that align with their tastes, offering both variety and relevance.

The front-end interface is clean and user-friendly, consisting of four main pages:

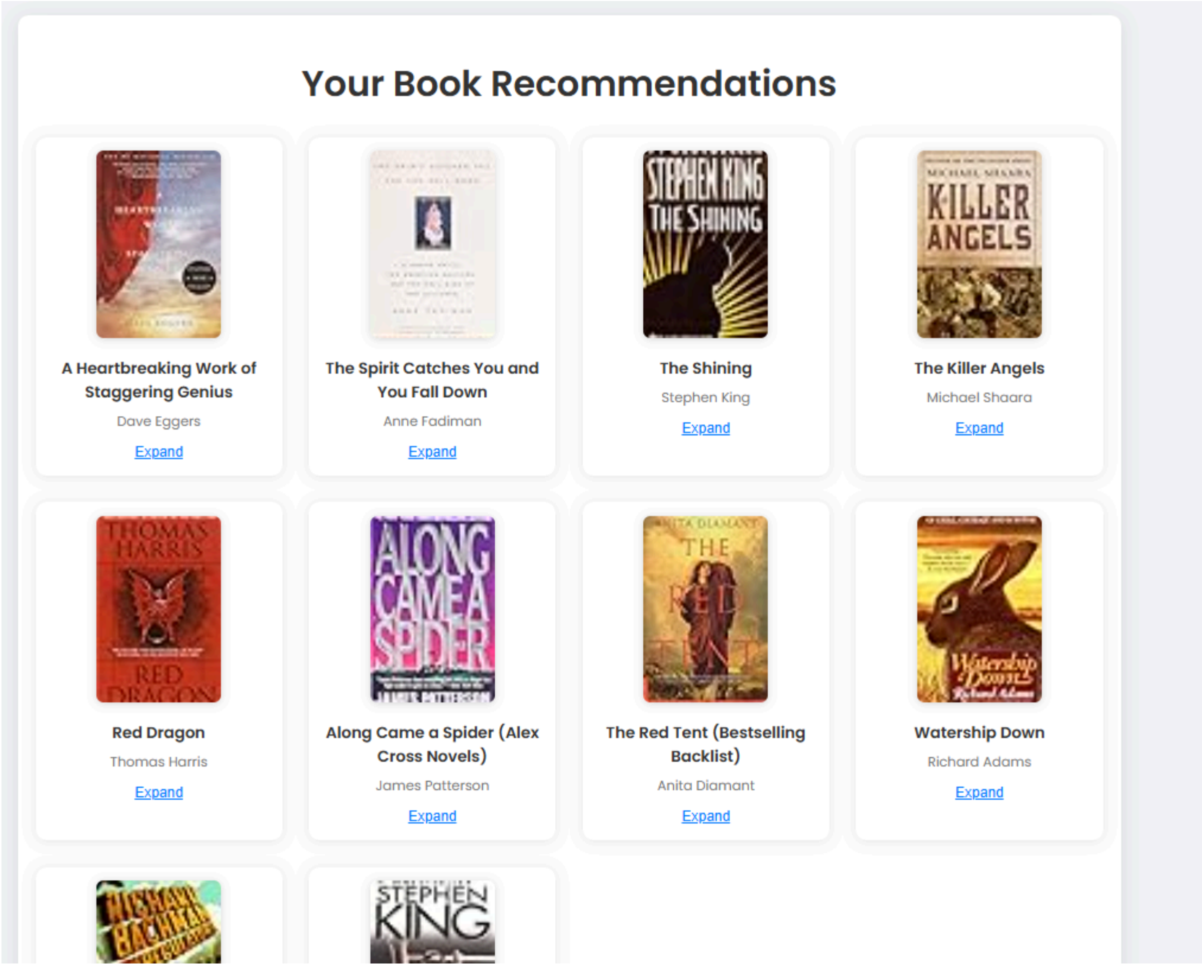
- **Sign-In Page**: Where the user inputs their name, age, and location to get started. In the background a user-ID is generated and the user is inserted in the dataset but without the name.
- **Book Selection Page**: Where the user can search for and select the books they wish to rate.
- **Rating Page**: Where the user rates the selected books on a scale from 1 to 5. Here, the user MUST rate the books they selected.
- **Recommendation Page**: Where the user receives a list of tailored book recommendations based on their ratings, preferences, and contextual data

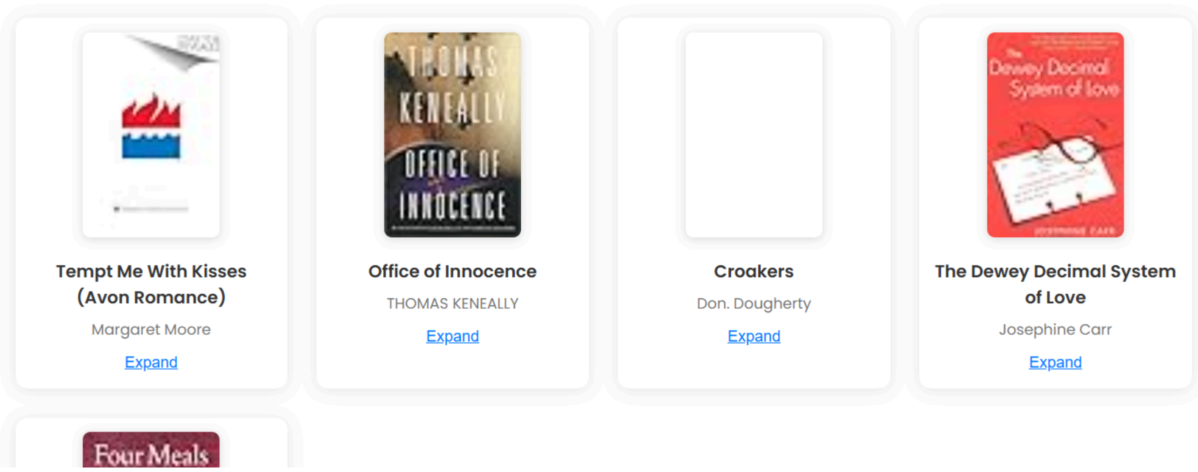
Overall, the front-end is designed to make the process seamless, engaging, and personalized, ensuring that users can quickly sign in, rate books, and receive relevant recommendations in just a few steps.

Recommendations examples:

1. User rates The Shining (Stephen King) with a maximum score of 5. Receives the following recommendations:
    a. Wolves of the Calla (The Dark Tower, Book 5) - Stephen King
    b. Valley of the Horses (Auel, Jean M. , Earth's Children.) - Jean M. Auel
    c. The Long Walk - Stephen King
    d. Shopgirl : A Novella - Steve Martin
    e. Rose Madder - Stephen King

2. User rates Peter Pan: The Original Story (Peter Pan) - J. M. Barrie with a maximum score of 5. Receives the following recommendations:
    a. Harry Potter and the Sorcerer's Stone (Book 1) - J. K. Rowling
    b. The Fellowship of the Ring (The Lord of the Rings, Part 1) - J. R. R. Tolkien
    c. Where the Red Fern Grows: The Story of Two Dogs and a Boy - Wilson Rawls
    d. Anne of Green Gables - L.M. Montgomery





4.2.4 Datasets

The 3 main dataset parts are Books, Users and Ratings. The Books dataset contains a pretty large number of 150k books with information like: ISBN, author, title and some URL for the cover picture. If it would be alone it would be a great dataset. The Users Dataset contains demographic and account-related information about the users interacting with the recommendation system. This dataset helps personalize recommendations and understand user behavior. Key columns typically include:

User-ID: A unique identifier for each user, used to link their activity and preferences.

Location: The geographical location of users, which can influence recommendations based on regional popularity.

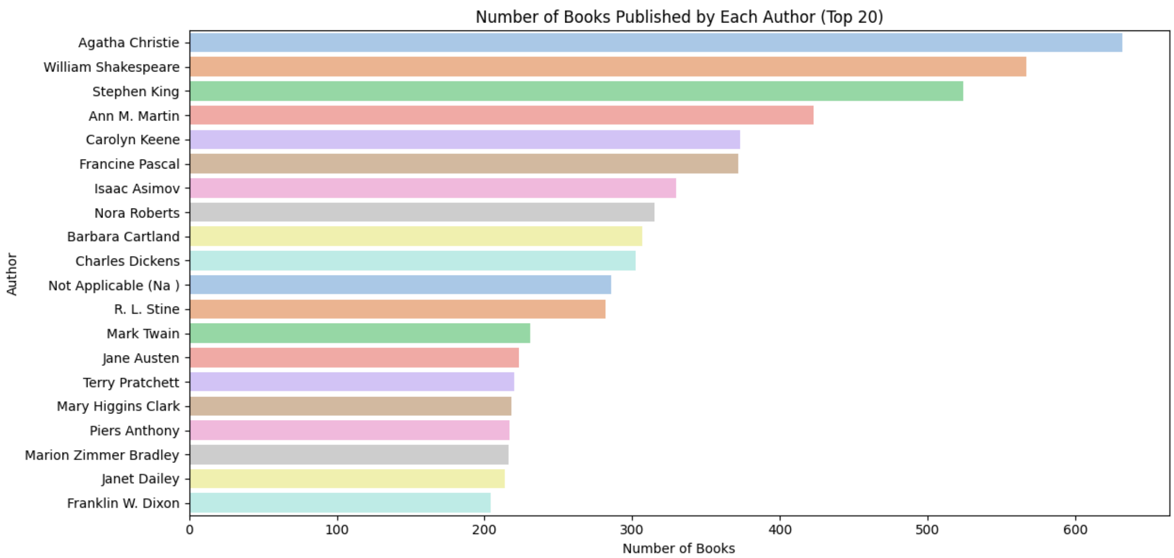
Age: User age, allowing the system to tailor recommendations to specific age groups.

This dataset allows the system to analyze user preferences and generate relevant recommendations, such as popular books in a user's region or age-appropriate content.

Although not explicitly mentioned, a Ratings Dataset is often used in recommendation systems. This dataset captures user feedback on books and could include:

- **User-ID**: Links ratings to specific users.
- **ISBN**: Links ratings to specific books.
- **Rating**: A numeric score representing how much a user liked a particular book.

This data is essential for collaborative filtering, where the system recommends books based on user ratings and preferences.



Problems with the datasets:

1. Even though the Books file contains a lot of information, the ratings file shows that just a small amount of books are rated and at the same time, a very small percentage of books are rated multiple times. There are 100k books but the ratings are around 25k (so just 25% of the books are rated in the maximum case) and not only that the 25k ratings are not even complete (a big percentage of them are rates of 0 which means the user has not read a specific book).

2. The users dataset even though has around 28k items for a better demographic and geographic understanding, it would need a more balanced and universal coverage. The users in the dataset are spread out in America, Spain, Portugal, UK, Russia and not all of the users submitted their location.

5. Economic analysis

What we have developed so far is a prototype for the book recommendation system. If this system were to be implemented on a professional scale, the economic analysis would follow a structure similar to this: The book recommendation system would aim to provide personalized and diverse suggestions to users, based on their reading habits, preferences, and trends. The economic analysis would define the target audience as readers seeking personalized book suggestions tailored to their unique tastes, including casual readers who need quick recommendations and avid readers exploring niche genres and authors.

As the user base grows, particularly in the initial years, the system's recommendation accuracy would improve through increased interaction data. The revenue model would include premium subscriptions for advanced features, targeted advertising, and possibly in-app purchases. Furthermore, partnerships with publishers and bookstores would offer additional revenue opportunities, and affiliate links to online book retailers such as Amazon and independent bookstores would provide a scalable, commission-based revenue model.

Costs

Development and Maintenance
The development phase includes designing and building the recommendation engine, user interface (UI), and mobile or web application. Initial costs for development and maintenance are projected to range between €260 000 and €300,000, covering salaries, software tools, and licenses. This is calculated for 1 year of development, enough time to get a working application with the desired functionality.

The core team required for development includes:

- Two Software Developers at approximately €44,000 each per year (Glassdoor, n.d.-a).
- Two Data Scientists specializing in machine learning, earning around €44,000 each annually (Glassdoor, n.d.-b).
- One UI/UX Designer at €33,000 per year (Glassdoor, n.d.-c).

The salary for the team is therefore estimated to be €242 000

Post-launch, ongoing development will be necessary to integrate updates, improve algorithms, and maintain the platform. Additional costs for these efforts are expected to stabilize at around €150,000 per year. The first year will be much more costly because we estimate that after the first year the need for developers will be significantly reduced.

Cloud infrastructure for hosting and data storage is a significant ongoing expense. Hosting on platforms such as AWS or Google Cloud is estimated at €1,000 to €5,000 per month, depending on the user base, equating to an annual cost of €12,000 to €60,000. Scalable cloud services are essential for managing growing user demand without performance degradation.

The projected range between €260 000 and €300,000 comes mostly from the salaries and the annual cloud compute cost. We estimate that the first year the cost of cloud computing will not be as high as €60,000. In the projected range there is also some additional cost from software tools and licenses. This is estimated to reach at least €6000, bringing the minimal cost to €260 000 with salary and cloud computing of €12 000.

5.1. Office and Equipment Costs

The development team will require office space and essential equipment for productivity. These costs are estimated as follows:

- Office Space: Renting a shared workspace or small office in Barcelona is estimated to cost €350 per month per employee, totaling approximately €21,000 annually for a team of 5.
- Internet: High-speed internet for the office is estimated at €70 per month, totaling €840 annually.
- Laptops and Monitors: High-performance laptops and additional monitors for development are expected to cost €2,000 per developer, resulting in a one-time cost of €10,000 for the entire team.

The total estimated cost for office and equipment for the first year is approximately €32,000

5.2. Marketing Costs

A robust marketing strategy is essential to attract and retain users. These costs are estimated as follows:

- Digital Marketing: Social media campaigns and Google Ads are expected to cost €2,500 per month, totaling €30,000 annually.
- Influencer Collaborations: Partnering with book influencers and bloggers is projected at €7,000 per year.

The total marketing budget for the first year is estimated at €37,000.

5.3. Compliance with GDPR and Privacy Regulations

Ensuring compliance with GDPR and maintaining data privacy involves legal consultations, audits, and infrastructure updates. This requires some legal consultations, which we estimate to cost €10,000 annually. The implementation of needed changes based on the consultations are not expected to be outside of the annual development costs we have already estimated.

Combining all estimated costs, the total for the first year ranges from **€339,000 to €379,000**, depending on cloud computing costs and other variable factors. These estimates include:

- Salaries: €242,000
- Cloud Infrastructure: €12,000 to €60,000
- Software Tools and Licenses: €6,000
- Office and Equipment: €32,000
- Marketing: €37,000
- GDPR Compliance: €10,000

6. Sustainability Analysis

1. Sustainability Goals Targeted by the Project

The primary sustainability goals of the book recommender system include:

- **Promoting Education**: By recommending books, the system encourages reading, which can foster knowledge and lifelong learning.
- **Reducing Waste**: A digital recommender system can minimize the production and waste of physical marketing materials, like flyers or catalogs.
- **Encouraging Access to Diverse Knowledge**: The system can promote inclusive education by suggesting diverse reading materials, contributing to cultural and social awareness.
- **Scalable and Sustainable Infrastructure**: Building a system that can scale efficiently while minimizing its environmental impact by optimizing cloud computing and server use.
- **Inclusivity and Accessibility**: Making reading accessible to all, regardless of socioeconomic background, by recommending free or affordable reading options, including libraries and e-books.

Relevant **Sustainable Development Goals (SDGs)**:

- **SDG 4**: Quality Education – The system can provide tailored recommendations that help users expand their knowledge base.
- **SDG 10**: Reduced Inequalities – Recommending diverse authors and perspectives can foster inclusivity and broaden horizons for readers.
- **SDG 12**: Responsible Consumption and Production – Promoting e-books or library services instead of purchasing new books could encourage sustainable consumption. Traditional book production has a significant environmental footprint due to the paper,

water, and energy used in printing, as well as transportation emissions. In this way we could help reduce deforestation and lower carbon emissions.

2. Sustainability Actions/Behaviors Undertaken by the Project

- **Cloud Hosting**: Utilizing energy-efficient cloud services, such as those powered by renewable energy sources (AWS, Google Cloud, etc.), can reduce the carbon footprint of running the app. Considerations should include using data centers that optimize cooling and power efficiency.
- **Server Optimization**: Implementing resource-efficient algorithms and ensuring that the system's architecture is optimized to scale dynamically with traffic (e.g., auto-scaling based on usage) minimizes wasteful resource consumption when demand is low.
- **Promoting Digital Access**: Recommending e-books, digital libraries, or audiobooks can reduce the carbon footprint associated with the production and transportation of physical books.
- **User-Centered Recommendations**: By recommending relevant content, the system avoids overwhelming users with unnecessary information, reducing digital clutter.
- **Environmental Awareness**: Recommending books on topics like sustainability, climate change, and environmental protection could encourage users to engage with ecological issues. This could be e.g. a separate section to not mess too much with the personalized recommendations.

3. Impact on Sustainability

- **Positive Impact**: The recommender system has the potential to positively contribute to sustainability in the following ways:
    - Encouraging continuous education and literacy, directly supporting **SDG 4**.
    - Fostering awareness of sustainable practices through book recommendations on ecological and social responsibility topics.
    - Reducing waste by encouraging digital reading and optimizing consumption behaviors.
    - Promoting inclusivity by recommending diverse voices in literature, aligning with **SDG 10**.
- **Potential Negative Impact**:
    - Running large-scale recommendation engines continuously can lead to significant energy consumption, especially as the user base grows.
    - If not optimized, the infrastructure could waste computational resources by keeping underutilized servers running, or running inefficient machine learning models that consume more energy than necessary.
    - If not designed ethically, the system could contribute to biased recommendations, limiting access to a broad range of perspectives.

4. Sustainability of the Proposed Solution

The proposed book recommender system can be sustainable if it:

- Utilizes efficient data processing and cloud services with lower carbon footprints.
- Actively promotes e-books or borrowing from libraries rather than encouraging the purchase of new physical books, contributing to **SDG 12**.
- Continuously evolves its recommendation algorithms to include diverse voices, promoting **SDG 10**.

However, sustainability could be compromised if the platform heavily relies on high-power servers without optimization or primarily promotes consumerism (e.g., focusing on commercial gains over meaningful content).

5. Ethical Considerations

- **Bias and Fairness**: The system must avoid bias in recommendations by ensuring that a wide range of authors, genres, and cultural perspectives are included.
- **Transparency**: Users should understand how recommendations are made and have control over their data and preferences.
- **Data Privacy**: The system must comply with data protection regulations (e.g., GDPR) to ensure that user data is handled ethically and responsibly.
- **Algorithm Accountability**: Developers must take responsibility for the algorithms used to prevent any harm caused by unintended biases or unethical recommendations.

By aligning with ethical AI principles and leveraging technology to promote education and inclusivity, this book recommendation system could have a lasting and positive impact on sustainability. We conclude that this system can be made sustainably.

7. Project management

7.1. Project Management Methodology and tools used

This sprint, we have continued working with Kanban after previously transitioning from Scrum. Kanban has proven to be a very effective method for our team, particularly because we don’t always have the opportunity to meet face-to-face. The flexibility of Kanban allows us to adapt to these circumstances while still maintaining a clear overview of our work.

We’ve been actively using the Kanban board to manage and prioritize tasks. This has helped us stay organized and aligned, ensuring that everyone knows what needs to be done and what the current priorities are.

Tools

To manage our backlog and user stories, we have continued to use Miro, which provides an easy and visual tool that works well for organizing tasks. The board we use is inspired by a Kanban board. *Figure 2* shows a screenshot of the board. Our user stores can later be broken down into developer tasks for insight. Using Miro, the tasks can also be given extra details. Some other platforms we discussed instead of Miro were Jira, but we found it a bit complex for our project's needs. A more basic approach would be simply a spreadsheet with task assignments, but this wouldn't be as visual as Miro.

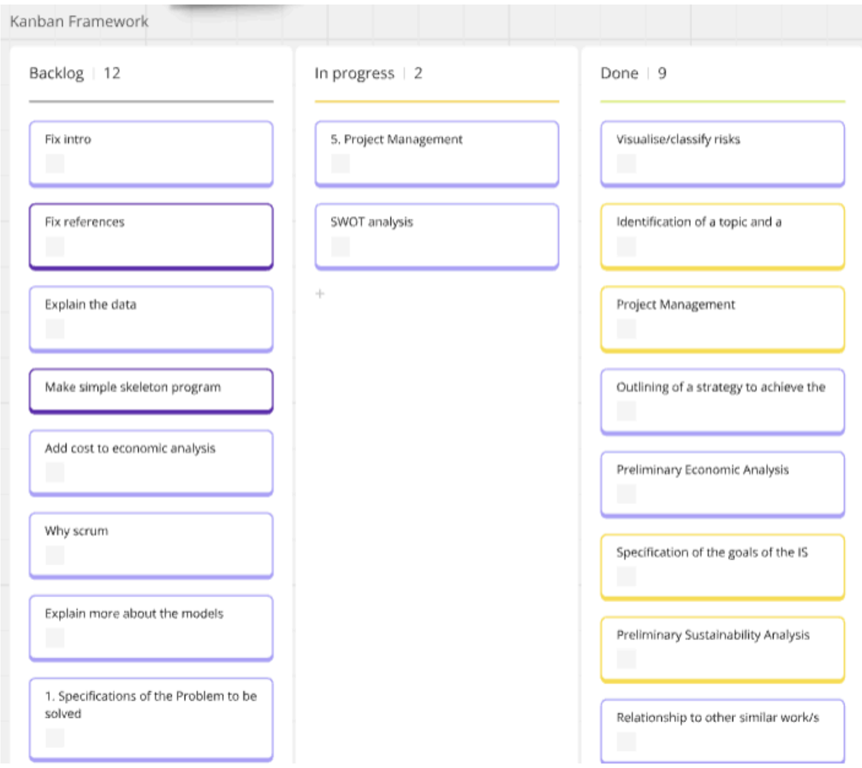


Figure 2: *A screenshot of our current board*

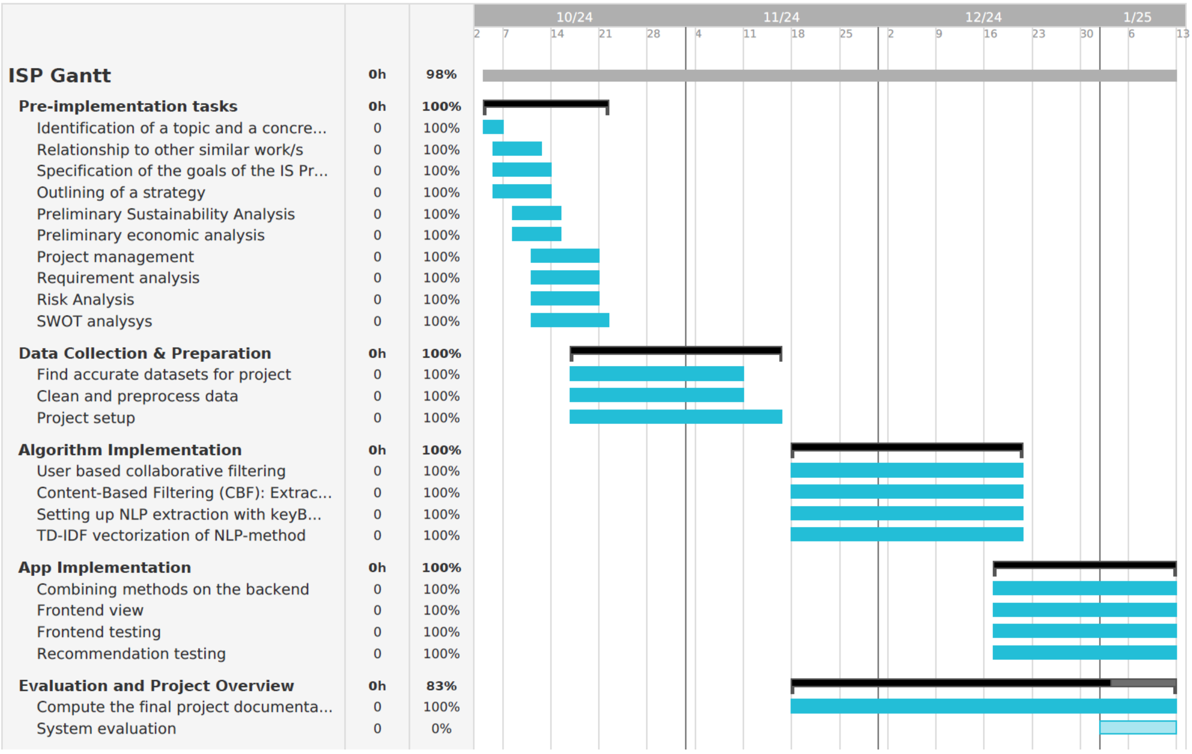
We use Github for development, since everyone in the group has good experiences with it, and it is one of the simplest platforms for using Git. Some other alternatives would be gitlab or azure, but the group has limited experience with those platforms.

We explored a couple of different programming languages for developing the project, including java and C# but we landed on Python. This is because of its simplicity and the fact that Python has the advantage of having an extensive library ecosystem. Python offers libraries like NumPy, Pandas, Scikit-Learn, and TensorFlow, which are particularly useful for building recommendation systems. It was also the developing language the group felt most comfortable with and had the most experience with.

Final IS Project Scheduling (Gantt's diagram)

The Gantt chart presented here is created after the final task assessment to provide a clear visual representation of the project's timeline and milestones. It outlines the various tasks we undertook throughout the development of the book recommendation system prototype, along with the specific timeline in which each task was completed. This chart serves as a valuable tool for tracking our progress and ensuring that all necessary steps were carried out in a structured and timely manner.

Unfortunately, due to time constraints, we were unable to conduct a formal evaluation of the system, as initially planned and displayed in the Gantt chart. While the prototype was developed according to the outlined tasks and timeline, the evaluation phase had to be deferred. However, this would have been an essential step in assessing the system's effectiveness, usability, and user satisfaction if we had more time.



Final Task Assignment

The model describes the initial task assignment for the project. This model does not include all future tasks, only the ones assigned up until this point. However this includes the tasks also done in the previous report. Here are both analysis, research and coding tasks shown.

Task	Lars	Aksel	Adiel	Elena	Task Status
T1-Identification of a topic and a concrete problem		x			Completed
T2- Relationship to other similar			x	x	Completed

work/s					
T3-Specification of the goals of the IS Project	x	x		x	Completed
T4-Outlining of a strategy			x	x	Completed
T5-Preliminary Sustainability Analysis		x			Completed
T6-6. Preliminary economic analysis		x			Completed
T7-Project management 1	x				Completed
T8-Specify problem			x	x	Completed
T9-Requirement analysis		x	x		Completed
T10-Project management 2	x	x	x	x	Completed
T11-Risk Analysis	x				Completed
T12-Find accurate datasets for project	x		x	x	Completed
T13-SWOT analysis	x				Completed
T14-Project setup	x	x	x	x	Completed
T16-User based collaborative filtering		x	x		Completed
T17-Content based filtering	x				Completed
T16-Setting up NLP extraction with keyBERT	x				Completed
T17-TF-IDF vectorization of NLP-method		x			Completed
T18-Combining methods on the backend	x	x	x	x	Completed
T19-Frontend view			x	x	Completed
T20-Frontend and backend testing			x	x	Completed
T21-Integrating methods			x	x	Completed

T22-Recommendation testing	x	x	x	x	Completed
T23-Finalise project documentation	x	x	x	x	Completed

Time sheet

Date	Task/ Subtask	Activity Description	Member involved	Time spent
Oct 10, 2024	T2, T3, T4, T13	Project prerequisites	Elena	6.5
14. okt. 2024	T1, T3	Initial work	Aksel	4
2 nov. 2024	T2	Relationship to other similar work/s	Adiell	5
3 nov. 2024	T4	Outlining a strategy	Adiell	2
4 nov. 2024	T8	Specify problem	Adiell	2
6 nov. 2024	T10	Project management 2	Lars	2
6 nov. 2024	T9	Requirement analysis	Adiell	4
6 nov. 2024	T12	Find accurate datasets for project	Adiell	1
Nov 6, 2024	T12	Find accurate datasets for project	Elena	1.5
Nov 10, 2024	T11	Risk Analysis	Lars	3
11 nov. 2024	T13	SWOT analysis	Lars	3
Nov 18, 2024	T14	Setting up the project, initial content and user based method	Lars	2
18. nov. 2024	T14	Setting up the project, initial content and planning	Aksel	5

Date	Task/ Subtask	Activity Description	Member involved	Time spent
22 nov. 2024	T14	Project setup	Adiell	2
Nov 22, 2024	T14	Project setup	Elena	2
Nov 25, 2024	T16	Implementing keyBERT and merging datasets	Lars	6
25. nov. 2024	T17	Start with TF-IDF implementation	Aksel	8
2. des. 2024	T17, T18	Further work on TF-IDF and NLP with the rest of the project	Aksel	7
Dec 20, 2024	T18	Combining NLP method with user based and collaborative filtering, using a hybrid method	Lars	7
2 ian. 2025	T18	Combining methods on the backend	Adiell	6
Jan 2, 2025	T18	Combining methods on the backend	Elena	8
4 ian. 2025	T19	Frontend view	Adiell	10
Jan 4, 2025	T19	Frontend view	Elena	6
8 ian. 2025	T20	Frontend and backend testing	Adiell	2
Jan 8, 2025	T20	Frontend and backend testing	Elena	2
9 ian. 2025	T21	Integrating methods	Adiell	4.5
9 ian. 2025	T22	Recommendation testing	Adiell	3
Jan 9, 2025	T21	Integrating methods	Elena	4.5
Jan 9, 2025	T22	Recommendation testing	Elena	2.5
10 ian. 2025	T23	Finalise project documentation	Adiell	0.5
Jan 11, 2025	T23	Working on documentation and presentation	Lars	8
11. jan. 2025	T23	Working on documentation and presentation	Aksel	15

Date	Task/ Subtask	Activity Description	Member involved	Time spent
Jan 11, 2025	T23	Finalise project documentation	Elena	6.5
Jan 15, 2025	T8, T10	Planning	Elena	2.5
Jan 19, 2025	T16, T17	Combining Content based and user based collaborative filtering	Lars	7

Total hours this iteration of the project

This does not include the work from previous iterations of the project

Teammember	Hours spent
Lars	38
Aksel	39
Elena	42
Adiell	45

8. User manual

8.1 Description of the Intelligent System's purpose

The book recommendation system we propose is designed not just to offer personalized book suggestions but to reignite the joy of reading in a world increasingly dominated by digital distractions. By leveraging advanced recommendation algorithms, our platform seeks to make reading more accessible, engaging, and tailored to individual preferences. What sets our project apart is its dual focus on personalization and inclusivity, ensuring users discover books they resonate with while also promoting diverse voices and perspectives.

8.2 Start up and shutdown

First of all, every library in the app.py should be already installed. After that, only the command: python app.py is needed to start the server. The first step is going to the browser to the link provided by the flask command and clicking on Get started. Next step is filling the user form with the necessary information and selecting next. The book selection part is where each user needs to select a list of books they read and with that selected list, they will rate each book in the following page. After this, the recommendations will appear. To close the app, in the terminal remove the process (CTRL + z or CTRL + c).

9. Conclusions and Future Work

At this stage, our project has achieved the development of a functional prototype that demonstrates the core concepts and potential of the book recommendation system. The application successfully integrates the three recommendation techniques, collaborative filtering, content-based filtering, and a NLP approach, showing how they work together to deliver personalized book suggestions.

There is still a lot needed to be done to make this a complete system. This is to a large extent just a prototype showing how the technology will work. The three different recommendation techniques are implemented, but they can still be improved. Most significantly, we should test the weighting between the different methods better. We believe another weighting would be ideal, and perhaps we could also use different weightings for different situations in a more complete system.

Despite its current functionality, the system remains a prototype and requires significant additional work to evolve into a complete, fully operational platform. A critical limitation is the inability to dynamically update user profiles or add new ratings, which restricts the system to static, preloaded data. Addressing this limitation is essential for enabling real-time personalization and ensuring that the system can adapt to users' evolving preferences, and is something that would be an urgent issue in further development.

The frontend part of the system is functional and shows what we want. To launch this as a finished product we would need to focus more on the design and usability of the frontend. It would be beneficial to conduct user tests, as the success of the application would depend quite heavily on the design. Even if the recommendations are good the users would still like a good frontend and usable website.

Lastly we should integrate with different suppliers of books. This could be online bookstores or libraries. That would allow the user to get their hands on the books and look at different places it is available. We could e.g. have a filtering function that only shows books available at a specific library that the user has chosen. This would make the application much more complete, and not be as dependent on other services to complete the experience of acquiring a book for the user.

References:

- The Guardian. (2024, November 5). *Fall in children reading for pleasure, report finds*. The Guardian. https://www.theguardian.com/books/2024/nov/05/report-fall-in-children-reading-for-pleasure-national-literacy-trust
- Glassdoor. (n.d.-a). *Software engineer salaries in Barcelona, Spain*. Retrieved November 16, 2024, from

https://www.glassdoor.com/Salaries/barcelona-spain-software-engineer-salary-SRCH_IL.0,15_IM1015_KO16,33.htm

- Glassdoor. (n.d.-b). *Data scientist salaries in Barcelona, Spain*. Retrieved November 16, 2024, from https://www.glassdoor.com/Salaries/barcelona-spain-data-scientist-salary-SRCH_IL.0,15_IM1015_KO16,30.htm

- Glassdoor. (n.d.-c). *User experience designer salaries in Barcelona, Spain*. Retrieved November 16, 2024, from https://www.glassdoor.com/Salaries/barcelona-spain-user-experience-designer-salary-SRCH_IL.0,15_IM1015_KO16,40.htm
- Mansurma, A., Kavitha, V., Abraham, J. M., & Benjamin, S. G. (2021). Survey paper on book recommendation system and comparative study on types of algorithms used. *International Journal of Creative Research Thoughts (IJCRT)*, *9*(6), 120–126. https://ijcrt.org/papers/IJCRT2106360.pdf

- Sarma, D., Mittra, T., & Hossain, M. S. (2021). Personalized book recommendation system using machine learning algorithm. *International Journal of Advanced Computer Science and Applications (IJACSA)*, *12*(1), 212–219. https://thesai.org/Downloads/Volume12No1/Paper_26-Personalized_Book_Recommendation_System.pdf

- A systematic review and research perspective on recommender systems https://www.semanticscholar.org/paper/A-systematic-review-and-research-perspective-on-Roy-Dutta/7f139ac3658ae3f8664f713b0f2af027fa990081

- The Guardian. (2024, November 5). *Fall in children reading for pleasure, report finds*. The Guardian. https://www.theguardian.com/books/2024/nov/05/report-fall-in-children-reading-for-pleasure-national-literacy-trust

- Aljohani, A., Alotaibi, M., & Alharbi, K. (2021). Personalized book recommendation system using hybrid filtering techniques. *International Journal of Advanced Computer Science and Applications (IJACSA)*, *12*(1), 227-232. https://thesai.org/Downloads/Volume12No1/Paper_26-Personalized_Book_Recommendation_System.pdf