Amir Ghari

NexLetter

A Newsletter Recommendation Engine

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor’s Thesis

Sep 2025

Abstract

Author(s): First Name Last Name

Title: Title of the Thesis

Number of Pages: xx pages + x appendices

Date: 28 August 2020

Degree: Name of the degree

Degree Programme: Name of the degree programme

Specialisation option: Name of the specialisation option

Instructor(s): First name Last name, Title (e.g., Project Manager)

First name Last name, Title (e.g., Principal Lecturer)

The text section of the abstract is written to fit the space used on the page. The text section uses the **Body Text No Spacing** style.

Keywords: Keyword

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The originality of this thesis has been checked using Turnitin Originality Check service.

**Contents**

[1 Introduction 1](#_Toc196307256)

[2 Theoretical Background 1](#_Toc196307257)

[2.1 Subheading 1](#_Toc196307258)

[2.2 Subheading 2](#_Toc196307259)

[3 Implementation 2](#_Toc196307260)

[3.1 Technology Stack Overview 2](#_Toc196307261)

[3.1.1 Backend Programming Language: Python 2](#_Toc196307262)

[3.1.2 Secondary subheading 3](#_Toc196307263)

[3.2 System Architecture 3](#_Toc196307264)

[3.3 Data Collection & Storage 3](#_Toc196307265)

[3.4 Recommendation Engine Design Data Collection & Storage 4](#_Toc196307266)

[4 Testing and evaluation 4](#_Toc196307267)

[4.1 How to add alternative text to images 5](#_Toc196307268)

[4.2 Subheading 6](#_Toc196307269)

[5 Conclusions 7](#_Toc196307270)

[4.3 Finish the document properties 7](#_Toc196307271)

[4.4 Check the accessibility of your thesis 7](#_Toc196307272)

[4.5 Save the Word document as an accessible PDF 8](#_Toc196307273)

[References 10](#_Toc196307274)

[Appendices 11](#_Toc196307275)

[Title of the Appendix 11](#_Toc196307276)

[Title of the Appendix 13](#_Toc196307277)

# Introduction

Write the introduction of your thesis here. Use line spacing 1.5 throughout the paper. Only the left edge is aligned, and the text is not hyphenated. Leave one blank line between paragraphs (press enter once).

Begin a new paragraph at the left margin, that is, do not indent the first line.

# Theoretical Background

## Subheading

There must always be text or a new subheading below each heading. Do not place a figure or table below a heading with no text in between. Label each figure and table appropriately. Provide a number, title and reference (if needed) below each figure and above each table.

Figure 1. Virtual studies completed by Metropolia students in the academic year 2009-2010.

There must always be text between a figure or table and a new heading.

## Subheading

There must always be text or a new subheading below each heading.

# Implementation

## Technology Stack Overview

This section provides an in-depth overview of the technologies utilized in the development of the newsletter recommendation engine. The selection of tools was driven by considerations of scalability, maintainability, and compatibility with content-based recommendation methodologies. The core stack comprises Python for backend development and data processing, PostgreSQL for structured data storage, and Next.js for frontend rendering.

### Backend Programming Language: Python

The backend system has been developed using **Python (version 3.x)** due to its extensive support for data manipulation, web development, and machine learning. Python is widely adopted in academic and industrial contexts for constructing recommender systems, primarily due to its rich ecosystem and ease of integration with APIs and databases.

Key libraries and frameworks utilized include:

* **Requests**: For handling HTTP requests to third-party services, including the newsdata.io API for fetching newsletter data.
* **Pandas** and **NumPy**: For efficient data cleaning, transformation, and statistical computation.
* **Scikit-learn** : To enable prototyping of recommendation algorithms such as TF-IDF-based similarity models or content clustering.

Python was selected over alternative languages (e.g., JavaScript/Node.js) for its proven effectiveness in building data-driven applications and its widespread use in academic projects involving recommendation engines [1][2][3].

### Database Management System: **PostgreSQL**

For data persistence, the system employs **PostgreSQL**, a powerful open-source relational database system renowned for its robustness, SQL compliance, and support for JSONB fields and advanced indexing mechanisms.

The database schema consists of the following core tables:

* newsletters (id, title, content, category, location, pub\_date)
* categories (id, name)
* locations (id, name)
* users (id, email, preferences)
* recommendations (newsletter\_id, user\_id, score)

PostgreSQL enables efficient query performance, particularly for operations involving keyword filtering, full-text search, and data aggregation. This makes it well-suited for content recommendation tasks where filtering by category, date, and region is essential [2][4].

### Frontend Framework: Next.js (Planned for Integration)

While the current scope of implementation emphasizes backend services and data pipelines, the frontend interface is planned to be developed using **Next.js**, a React-based framework that supports server-side rendering (SSR) and static site generation (SSG). These capabilities are critical for delivering a performant and SEO-friendly user experience when displaying categorized newsletters and recommendations.

Next.js offers native support for building API routes, which simplifies the integration between the frontend and backend without the need for separate middleware services. It has been successfully adopted in similar projects involving content aggregation and news delivery platforms [3][5].

### Development Infrastructure and Tools

To streamline development and deployment, the following tools and services are incorporated into the workflow:

* **Git and GitHub**: For version control, team collaboration, and continuous integration.
* **Docker** *(optional)*: For containerizing the application to ensure environment consistency across development and production stages.
* **Vercel** or **DigitalOcean** *(optional)*: Intended for frontend and backend deployment, respectively, offering scalable infrastructure solutions.

These tools enable modular development and facilitate rapid iteration during the engineering cycle. Moreover, the use of Docker supports reproducibility, which is essential for both academic validation and production reliability [2][4].

## System Architecture

There must always be text or a new subheading below each heading.

Quotes use the **Quote** style. The paragraph containing the citation passage (immediately before the citation) uses the **Normal Text Before a Quote or List** style to leave a shorter paragraph spacing between the citation and the passage.

A multi-line direct quote is written in font size 12. The text uses line spacing 1, and the text is indented. Direct quoting uses the **Quote** style of the template. A citation is given in the quotation.

After indentation, the text continues from the left edge in the body text style.

## Data Collection & Storage

A list in the text uses the **List** style. The paragraph before a list uses the **Normal text before a quote or list** style.

This is the first item in the list.

The second item of the list here contains a long text that spans multiple lines. The left edge aligns automatically.

This is the first item in the list.

The fourth item in the list is here.

When the list items are sentences, they begin with a capitalized letter, and the list items end in a period.

When the list items are not sentences, they begin with a lowercase letter, and the last list item ends in a period. The thesis consists of

words

clauses

sentences

paragraphs

chapters.

## Recommendation Engine Design Data Collection & Storage

# Testing and evaluation

According to accessibility requirements, images must have alternative text. Alternative text is not the same thing as a caption. Alternative text is a description of the content of an image read aloud by screen readers used by the visually impaired. It is not advisable to repeat the caption in the alternative text because screen readers read both contents.

When writing alternative text, think about what information you will not receive if you do not see the image. Use short sentences and plain language. Tell the essential about the picture - you don't have to explain everything.

Alt text can be blank for a decorative image. Decorative images are images that do not convey any information or that have been added for layout.



Figure 2. Long-tailed jaeger is common in Finnmark's mountain plateau in northern Norway. Its main wintering site is in the South Atlantic west of Africa.

## How to add alternative text to images

An alternative text is given to an image in a Word document as follows (Office 365 version):

1. Move the cursor over the image and right-click.
2. Select “Edit Alt Text…” to open the Alternative Text window.
3. Write a brief explanation of the essential content of the image.
4. If the image is purely decorative or the caption contains all the relevant information, mark the image as decorative.

Graphical user interface, text, application

Description automatically generated

The Microsoft Office 2016 version of Word is slightly different:

1. Move the cursor over the image and right-click.
2. Select “Format Picture…”
3. In the “Format Picture” window, select the third icon “Layout and Properties”.
4. Select “Alt Text” and enter a description of the image content in “Description”. Do not write anything under “Title”.

## Subheading

There must always be text or a new subheading below each heading.

Graphical user interface, text, application

Description automatically generated

# 5 Conclusions

## Finish the document properties

Once the content of your thesis is in order, finalise the document by specifying its properties. It is essential to ensure that the pdf file is accessible when you convert a Word file to PDF format. Type a title for the document in the File menu, under Info. Enter the title of your thesis as the title.

Graphical user interface, text, application

Description automatically generated

## Check the accessibility of your thesis

Word has a feature that lets you check the accessibility of a document.

1. On the File menu, click Info.
2. Then click Check for Issues.
3. Click Check Accessibility.

Graphical user interface, application, Word

Description automatically generated

The Accessibility Check window will then appear on the right side of the Word. The results of the scan show possible errors and warnings. For more information about results, click the item name in the results list. Word also tells you the reason for the repair, as well as give repair instructions. At least fix any errors.

## Save the Word document as an accessible PDF

Once you have checked your thesis for accessibility, convert it into an accessible PDF document.

1. Create a PDF file using either the Export function (Create PDF) or the Save As function.
2. In the save options, select **Document properties** and **Document structure tags for accessibility**.
3. Click Create bookmarks using Headings.

Graphical user interface, text, application, email

Description automatically generated

Do not use the Print to PDF function because the result is not accessible PDF.

References

1. Neamtu, R. (2023). *Major Qualifying Project*. Worcester Polytechnic Institute. Retrieved from <https://digital.wpi.edu/downloads/2b88qg37v>
2. Phan, T. D. (2022). *Futures Platform Web Application*. Theseus.fi. Retrieved from <https://www.theseus.fi/bitstream/handle/10024/750360/ThesisReport_PhanDat_E1800932.pdf>
3. Rutuja, B., & Nalinipriya, G. (2025). *Developing a Sophisticated Collaborative Filtering-Based Web Application*. Taylor & Francis. Retrieved from <https://www.taylorfrancis.com/chapters/edit/10.1201/9781003501244-81>
4. Suva, A. (2022). *Design and Development of CoPAS Marketplace System*. Masaryk University. Retrieved from <https://is.muni.cz/th/eoxdj/thesis_Archive.pdf>
5. Zhao, Z. (2023). *Building a Live News Application with Next.js*. Theseus.fi. Retrieved from <https://www.theseus.fi/bitstream/handle/10024/793229/Zhao_Ziang.pdf?sequence=2>

Appendices

Title of the Appendix

The content of the appendix is ​​written here. Below are instructions for adding and removing attachments so that the headers remain the correct.

1. Instructions for adding a new attachment:
2. Move the cursor to the end of the last existing attachment page.
3. Choose the Page Layout tab. From the Page Break ribbon select Next Page under Section Breaks. This completes the printing of the new attachment, but the number in its header is not correct.
4. Double tap the header of the new attachment page with the wrong attachment number. If the “Link to previous” option is selected in the ribbon, press that button so that the option is no longer selected.
5. Please correct the attachment number.

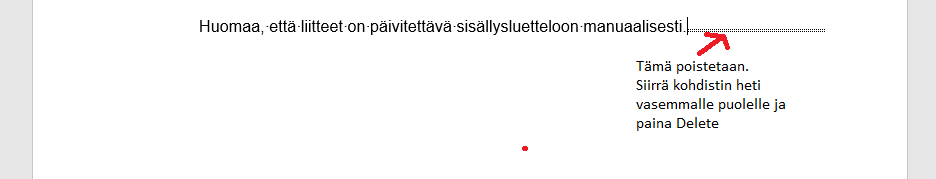
Instructions for removing an unnecessary attachment:

1. First select the entire attached page and press Delete to delete its contents.
2. When you are at the beginning of the attachment page you have emptied (see figure), double tap the header of the blank attachment page and press the Link to Previous button on the ribbon. The following dialogue box appears:



Answer Yes.

1. From the Home tab, toggle hidden characters if they are not visible:Piilomerkki painikkeen kuvake.
2. Remove the section break before the unnecessary attachment (see figure below).



Title of the Appendix

Content of the appendix is placed here