AI-Synthesized Solo Podcast Platform For Knowledge Enhancement

Prof. Smita Dashpute[1], Saumya Yadav[2], Dhwani Tiwari[3], Reecha K. Vasan[4]
[1, 2, 3, 4] K.C. College of Engineering & Management Studies & Research

Abstract- *Digital content creation has seen a massive surge, yet solo creators often face technical hurdles in scriptwriting and professional audio production. This paper presents 'Podcastr', an innovative platform designed to automate the end-to-end podcasting lifecycle. Unlike traditional systems that rely on manual recording, our proposed system integrates the Groq LPU (Language Processing Unit) for near-instant script generation using the Llama 3 model. For audio synthesis, we utilize the Microsoft Edge TTS engine to deliver human-like neural voices, eliminating the need for expensive studio equipment. The backend is architected using FastAPI to ensure asynchronous processing of AI tasks, while Hotwire Turbo is implemented to maintain persistent audio playback during site navigation. Our results demonstrate a significant reduction in production time—from hours to mere seconds—making professional podcasting accessible to individual knowledge-sharers.*

Keywords: AI-Driven Content Creation, Asynchronous Web Frameworks, Edge-TTS, FastAPI, Groq Llama 3, Hotwire Turbo, Natural Language Processing (NLP), Neural Speech Synthesis, Persistent UI, Razorpay Integration, Solo Creator Economy, Text- to-Speech (TTS).

I. INTRODUCTION

People really love listening to podcasts these days. The reason is that podcasts are a way to learn new things while you are doing other stuff. You can just put on your earphones. Listen to a podcast while you are traveling or exercising or even working at home. Listening to podcasts is a lot better than reading papers or watching videos. The thing is that making your podcast is not easy. Lots of people want to share what they know about podcasts. They do not because the technical part of making a podcast is very hard and expensive for one person to handle.

If you want to start your podcast today, you will have to deal with a lot of problems with your podcast. First you need a script for your podcast. Writing a script for your podcast takes a lot of time. Then you need a room and a good microphone for your podcast or the sound of your podcast will not be good. After that you have to learn how to edit the audio of your podcast, which's not easy for everyone to do with their podcast. Because of these problems with making a podcast many people with ideas for a podcast just give up on their podcast. We made Podcastto solve this problem with making a podcast.

Our project Podcast is like a studio that you can use with one click for your podcast. All you have to do is give us a topic for your podcast and our system will do all the work for your podcast. We used FastAPI for the part of the podcast that people do not see because it is very fast for our podcast. The important part of Podcasts is the Groq Llama 3 model, which can write a script for your podcast in one second for your podcast. We also used Edge TTS to make a voice that sounds like a person for your podcast so you do not even need a microphone to make a podcast with Podcastr. To make sure that the audio of your podcast keeps playing when you click on links, for your podcast we used Hotwire Turbo for your podcast. This paper will show you how we used these tools to make it easy and free for anyone to make a podcast with Podcastr.

II. LITERATURE REVIEW AND COMPARITIVE ANALYSIS

The development of Podcast involved studying existing technologies. Where they fall short for solo creators. Our research focused on four areas:

1. Evolution of Generative AI: We looked at how AI has changed over time from simple text completion to generating long scripts that sound human. Models like Llama 3 are an improvement. We studied how fast these models can respond. Old APIs took minutes to respond, which's too slow.. With Groq’s technology we can create content almost instantly. This is a step forward.
2. Neural Speech Synthesis: Old text-to-speech systems sounded robotic. Were hard to listen to.. New neural networks, like Microsoft’s Edge TTS have changed that. These models understand the rhythm and intonation of speech. Our study showed that Edge TTS provides quality audio and fast server response. This makes it perfect for a real-time podcasting service.

3. Modern Web Architectures: Developers used to have to choose between websites and complicated single-page applications. Our review found that a server-rendered monolith using FastAPI is better for media platforms. When combined with Hotwire Turbo it provides an stable experience. This is important for keeping the player working smoothly across page transitions.
4. Gap in Existing Platforms: There are three types of platforms there but none of them fully solve the problem for solo creators. Hosting sites like Anchor host audio; they don't help with creation. Editing tools like Descript help with editing. They don't provide hosting or monetization. E-learning sites like Udemy are built for video. Don't support audio-first learning.

Podcast fills this gap. It is a platform that handles production, seamless playback and revenue generation all in one dashboard. This makes it a unique end-, to- end solution that doesn't exist in the market. Podcast uses Groq and Edge TTS for production. It provides playback with Turbo. It also offers revenue generation with Razorpay. Podcast is designed to meet the needs of creators.

III. PROBLEM STATEMENT & OBJECTIVES

The world of podcasting is a mess. If you are a solo educator you have to use a lot of tools just to make one episode and that is a huge problem. We found four issues that our project, Podcast tries to fix:

A. PROBLEM STATEMENT

High Production Stress and Costs: The biggest problem is how long it takes to make just one audio file. Between writing, recording and editing out sounds like "umm" and "ahh" a ten-minute podcast can take five hours to finish. Most teachers or experts are not engineers and they do not have the money to hire one. This means many great educational ideas are never shared because the process is too tiring for the educators.

The Audio Disruption Problem: Have you ever noticed how most websites reload everything when you click a link? On a podcast site this is a disaster because the audio stops away. This breaks the flow of learning for the listener. We wanted a system where the audio player stays active at the bottom of the page no matter where the user clicks on the site so the podcast, keeps playing.

Content Trapped in One Language: solo creators only make content in their native language. They do not have the tools to translate their work so they lose out on millions of listeners who want to listen to the podcast Podcastr. There is a need for a platform that can automatically translate and re-synthesize audio into different languages like Hindi or Spanish so the podcast, Podcast can, reach more people.

Hard to Monetize for Beginners: systems like YouTube or Spotify ads only pay well if you have millions of hits. Niche educators who have a serious audience do not have a simple way to offer subscriptions or take direct payments, which makes it hard for them to keep their project the podcast, running financially.

B. OBJECTIVES

To fix these things our project, Podcast-focused on
:

- Automating the content pipeline using Groq and Edge TTS so no manual recording is needed for the podcast, Podcastr.
- Keeping the UI persistent using Hotwire Turbo within FastAPI so the audio of the podcast Podcastr, never breaks.
- Adding Reach through AI-based translation modules so the podcast, Podcast can be heard by people all, over the world.
- Secure Payments using Razorpay so creators of the podcast Podcastr, can actually earn money.
- Auto-Healing features that automatically fix or re-generate any missing files using saved metadata so the podcast Podcastr, always sounds good.

IV. SYSTEM ANALYSIS & DESIGN

The design of Podcast is focused on creating a bridge between AI models and a simple user-friendly interface. We wanted to make sure that even a person with no knowledge could produce a professional podcast. This section breaks down how we organised the Podcast system to achieve this.

A. System Overview

At its core Podcast is an automation platform that handles the entire lifecycle of an audio episode. The Podcast system does not just host files it actually creates them. We analyzed the needs of creators and realized that their biggest problem is the time spent on manual tasks. Therefore, our overview focuses on a Topic-to-Audio flow. The Podcast system is divided into two environments: the Creator Dashboard, where scripts are generated and edited and the

Listener Arena, where users can browse, search and listen to content through a persistent non-interruptive player. By using a Server-Side Rendered approach with FastAPI we ensured that the Podcast application remains lightweight and fast for both roles.

B. System Architecture

Our Podcast architecture follows a pattern to ensure that the AI processing does not block the main web server. We used a Three-Tier Architecture:

- Presentation Layer: Built with Tailwind CSS and Jinja2 this layer handles the look and feel. It uses Hotwire Turbo to manage page transitions without breaking the stream.
- Application Layer: This is the brain of the Podcast project, powered by FastAPI. It manages the logic for authentication talks to the Groq Llama 3 API for scripting and triggers the Edge TTS engine for voice synthesis.
- Data Layer: We used SQLAlchemy with SQLite for storing data. This includes everything from user profiles and hashed passwords to the metadata for every generated podcast. This layer also supports our Auto-Healing logic, which can recreate missing files using the stored text.

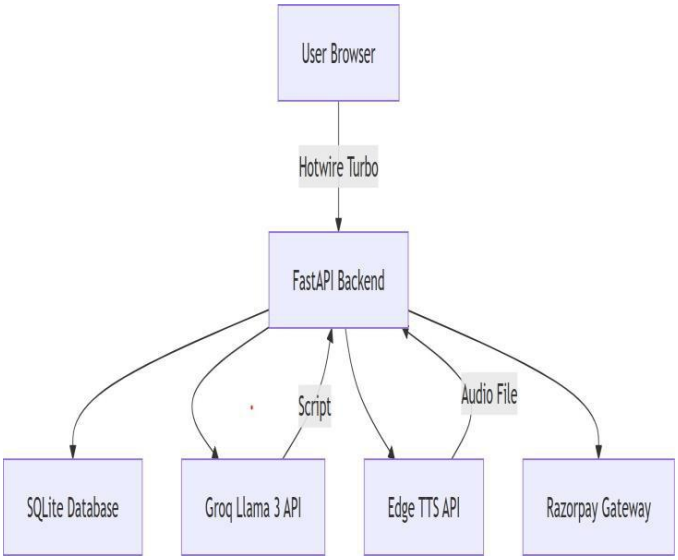

Figure- 1: System Architecture

C. Functional Design

The functional design of Podcast is centered around automation and reliability. One of the functions we designed is the Asynchronous Task Queue. When a creator requests a podcast the Podcast system does not make the user wait on a loading screen. Instead it starts the generation in the background.

- Scripting Logic: The Podcast system sends a prompt to the Groq LPU, which returns a clean formatted script.
- Audio Synthesis: This script is then fed into the Edge TTS module to produce a high-fidelity MP3.
- Persistent Navigation: This is a functional design choice where we used Hotwire Turbo to ensure that the global audio player remains active even when the user is moving between the Trending and Profile pages.

D. System Workflow

1. The workflow of Podcast is set up as a step- by-step process that automates the process of creating audio. Our main goal is to make things easier for the creators. If you look at the workflow diagram you can see that the process has a stages it goes through.
2. Initial Creator Input: The process starts when the Solo Creator types in a topic idea into their dashboard. This topic idea is the starting point for the automated process. You can see the steps in the workflow diagram.
3. Asynchronous Script Generation: When the Solo Creator types in the topic idea it gets sent to the FastAPI backend away. The FastAPI backend does not stop the user interface from working. Instead it sends a message to the Groq LPU to generate a script. The script is generated one part at a time.
4. Data Persistence: When we get the script from Groq we save it in the SQLite database. We save the creators name, the topic and the time it was created along with the script. This way we can make sure all the data is safe.
5. Simultaneous Audio Synthesis: Right after we save the script the system turns on the Edge TTS engine. The Edge TTS engine uses the script to create the audio file. The audio file sounds like it was made by a human.
6. Publishing to Library: When the audio file is ready we add it to the library. For the people listening the process is easy. They can look through the podcasts search for something and play it. The podcast will keep playing without stopping because of the way Hotwire Turbo works.
7. Monetization Upgrade Flow: The system is also set up to handle business things. If a listener wants to upgrade to the Pro plan the backend starts a checkout

process using the Razorpay Gateway. After the payment is verified the Pro features, like translations are

unlocked right away on the creators dashboard. The Podcast system makes it easy for the creators to use these features.

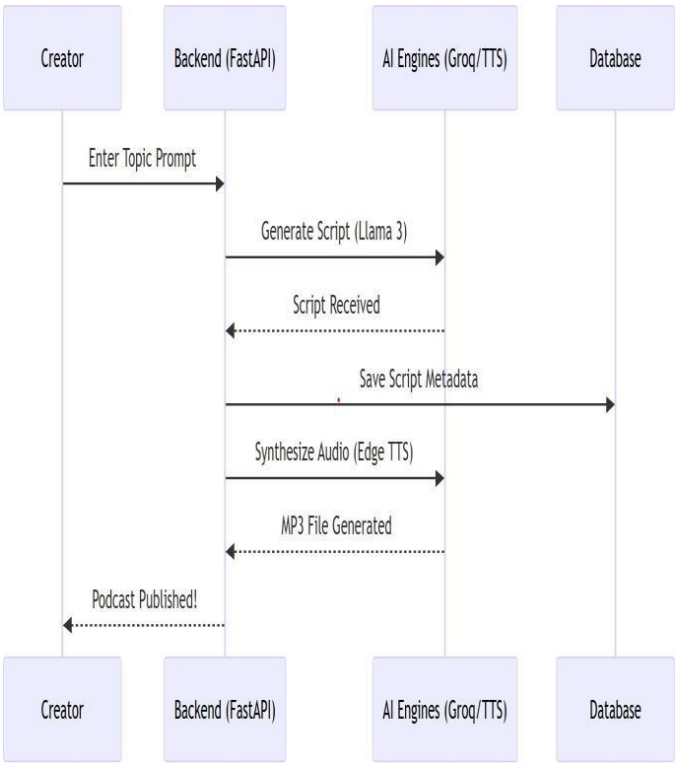

Figure- 2: System work flow Diagram

V. METHODOLOGY

The podcast is a platform that helps users to generate, listen, and manage podcasts easily under one roof. For achieving this we follow a step-by-step method which includes understanding what users need, designing the system, adding AI features, and testing everything to make sure it works smoothly.

First the system allows users to create an account and log in securely. All the details entered by the users including basic details like email, name, and password are stored safely using secure login methods so that only authorized users can access their data in the system. The password is securely stored using Passlib and Bcrypt, and secure login is handled using JWT (Python-JOSE). After securely logging in users can create a podcast by entering a specific topic. The system then sends this topic to an AI service (Groq API – Llama 3) which then generates a complete podcast script automatically. Next that generated script is converted into audio using a Text-to-Speech service (Edge TTS). This creates a realistic .mp3 audio file which is afterwards saved in the system. All the user details, scripts and audio files is then stored in the database. The podcast is afterwards shown on the screen with an audio player, where users can listen and read the script at the same time.

The system provides a smooth navigation using Hotwire Turbo, so that all the pages load without full refresh and the audio keeps playing continuously. Users can also search and discover podcasts, create playlists, and translate scripts into different languages using AI. The platform includes a payment system using Razor pay API, where users can upgrade to a Pro plan. Payments are verified securely. Users can also track themselves on the dashboard, where they can view their podcasts, manage their playlists and check their subscription status as well.

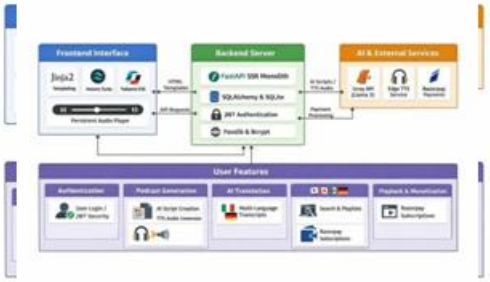

Figure- 3: Methodology Block Diagram

Figure 3 represents and states the overall system architecture and data flow of this podcast. The process starts with user input (topic), which goes through the frontend to the backend. The backend processes the request, interacts with the database, and connects with external services like Groq API, Edge TTS, and Razorpay. The final output (script and audio) is sent back to the user interface and displayed as a playable podcast.

VI. IMPLEMENTATION & RESULTS

Implementation The AI Podcast is a platform built in a efficient way for users to easily create and listen to podcasts under one roof. The system is developed and built using FastAPI for the backend which handles all the main logic, database operations, and page rendering. Instead of using a separate frontend framework, the system uses server-side rendering with templates. To give users a smooth experience like a modern app, Hotwire Turbo is used, which allows users to experience update without fully reloading, so the app feels

faster, polished and more interactive. The audio player is designed and built to stay active even when users move between pages, so that they can continue listening without any interruption. SQL Alchemy manages the database in order to store user data, podcasts, user details and scripts. AI integration helps and allows users to either write their own script or generate one using AI. Large Language Models (LLMs) are used in order to create podcast scripts based on user inputs. After generating the script, the Text-to-Speech (TTS) technology then converts the text into audio automatically.

The platform also includes features like:

- Discover section to explore trending podcasts
- Playlist creation for saving favorite content
- Transcription view to read podcast content
- Multi-language translation for better accessibility

The subscription system allows the users to choose between free and paid plans based on their needs. The system is designed and built to be simple, fast, efficient and user-friendly while using advanced AI features.

B. Results

The AI Podcast system shows strong improvement in making and building a podcast easily and fast compared to other traditional methods. Users can create a full podcast (script + audio) in just a few steps which saves a lot of time and effort.

Figure 4 :Landing Page

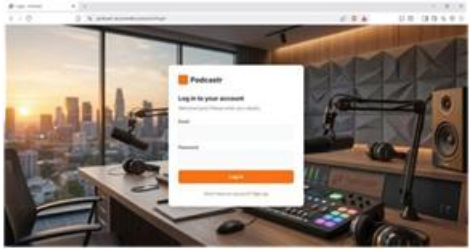

Figure 5 : Log in Page

The AI-generated scripts not only help users who do not have writing skills but also ideas. The Text-to- Speech feature successfully converts scripts into natural-sounding audio which makes the platform even more useful for beginners and professionals. The smooth navigation which is built using Hotwire Turbo improves user experience, as the pages loads faster and feel like a mobile app. The persistent audio player allows continuous listening, which increases user engagement.

Figure 6 : Home Page

Figure 7 : Podcast Interface

Features like transcription and translation not only makes the platform more accessible for different kind of users but also improves understanding of content. The discover and playlist features help users find and organize podcasts easily. The subscription model allows users to choose between free and upgrade when needed flexibly Overall, the implementation of this system proves that combining AI with a simple and efficient design can make podcast creation more efficient, smooth, interactive, motivating and accessible. It provides the users with modern solutions which is the very essence of making this system.

Figure 8 : Podcast Audio Generation

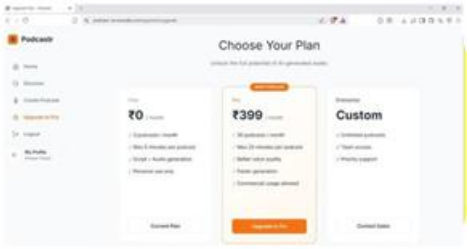

Figure 9 : Subscription page

VII. FUTURE SCOPE

We created Podcast to make content for everyone and we think this is just the start. When building Podcast we found areas where it can grow and improve.

- Personalized Voice Cloning: Our Edge-TTS gives voices but the next big thing is to let creators clone their own voice. This way the audio generated by Podcast will sound like the creators voice. They can keep their touch and emotional bond with listeners. Creators will have their voice with Podcastr. Podcast will help creators clone their voice.
- Intelligent Background Scoring: Music makes a podcast great. We want to make a system that checks the mood of the script and makes a custom background score for free. This way creators won't have to find and sync music tracks themselves. The podcast will have its music system that works well. Podcast will make it easy for creators to add music.
- AI-Driven Video Avatars: Video platforms are popular so we want to make Podcast more than audio. We can add talking-head models to turn a podcast into a video podcast with an avatar that speaks the content. This is perfect for YouTube and Instagram, where people watch videos. Podcast can be used on these platforms easily. Podcast will help creators make video podcasts.

Real-Time Interactive Podcasts: We want to make podcasts a two-way conversation where the listener and the podcast can talk to each other. Imagine a listener asking a question

- on their phone and the system answering instantly. This will make podcasts more engaging and fun to listen to. Podcast will make podcasts interactive.
- Multi-Language Dubbing: We can translate words now. We want to make dubbing perfect so that the podcast can reach more people. The AI should not just translate words. Also match the tone and expression of the original content, in many languages. This way Podcast can be used by people who speak languages and they can enjoy the content too. Podcast will help people enjoy content in their language.

VIII. CONCLUSION

When we first started working on Podcast our main goal was simple: we wanted to see if we could make the voice of a solo creator powerful as a full production team. After months of building and testing Podcast this project has proved that technology can indeed level the playing field. We realized that the real reason most people do not share their knowledge is not a lack of ideas but the exhausting technical work that comes with podcasting. By automating the scriptwriting and audio synthesis phases of Podcast we have successfully removed those barriers.

The integration of FastAPI and Groq allowed us to achieve speeds with Podcast that we did not think were possible at the start.. More than the speed of Podcast it is the user experience that matters most. By using

Hotwire Turbo with Podcast we fixed the problem of audio cutting out during navigation. A small change that makes a huge difference for the listener of Podcastr.

In conclusion Podcast is more than a piece of software it is a solution for the silent expert who has a lot to say but does not have the time or money to say it. We believe that by making creation with Podcasts easy as typing a topic we are opening the door for a new wave of global educators. This project is the beginning of our journey to make digital storytelling with Podcast accessible, affordable and truly inclusive, for everyone everywhere.

We have successfully built and completed this project under the guidance of Prof. Smita Dashpute.

REFERENCES

[1] Meta Platforms Inc., "Llama-3: Developing the generation of open-foundation chat models " Meta AI Research Reports, Menlo Park, CA 2024. [Online]. Available: https://ai.meta.com/llama/ I really think Meta Platforms Inc. Is doing something by sharing Llama-3 with the open-source community. This is a deal for developers like us. Meta Platforms Inc. Is changing the game.

[2] Groq Technologies, "Breaking the speed barrier: LPU architectures for lightning-fast AI inference " Technical White Paper, Mountain View, 2024. I read this paper by Groq Technologies. I was blown away by their LPU

speed. It is way faster than GPUs. Groq Technologies is really pushing the limits.

[3] S.. P. Norvig, Artificial Intelligence: A Modern Approach, 4th Global Edition. Pearson Education, 2020. This book by S.. P. Norvig is my go-to book when I need to learn about AI basics. I get stuck sometimes and this book helps me out.

[4] I. Goodfellow, Y. Bengio and A. Courville, Deep Learning (Adaptive Computation and Machine Learning series). MIT Press, 2016. If you want to understand networks, you should read this book by I. Goodfellow, Y. Bengio and A. Courville. It is the book for learning about networks.

[5] T. Tirollo, "A study on Asynchronous Frameworks: Why we chose FastAPI over Flask " Journal of Software Engineering vol. 14 Pp. 112–128, May 2021. This paper by T. Tirollo helped me decide to use FastAPI for our projects. We need real-time results and FastAPI is the way to go.

[6] Microsoft Corp., "Neural Speech Synthesis: How Edge-TTS is changing the accessibility landscape " Azure Cognitive Services Documentation, 2023. Microsoft Corp. Is making progress in making AI voices sound human. This is a step for accessibility.

[7] D. H. Hansson "The Hotwire Manifesto: web apps without the JavaScript fatigue " Basecamp Engineering Journal, 2021. Available: https://hotwired.dev/ I love this approach by D. H. Hansson because it keeps things simple. I like solutions.

[8] R. Fielding, "Architectural Styles and the Design of Network-based Software " Ph.D. Dissertation, University of California, Irvine, 2000. This is a paper by R. Fielding. It is a bit long. It explains the web really well.

[9] J. Zenich, "Managing Asynchronous Python: Real- world performance at scale " in Proc. 12Th International Python Developers Conference (PyCon) 2022 pp. 45–52. Dealing with async tasks in Python can be tough. This paper by J. Zenich was really helpful.

[10] A. Vaswani et al. "Attention is all you need: The transformer revolution," in Proc. 31St Conf. Neural Information Processing Systems (NIPS) 2017. This paper by A. Vaswani started it all. Every LLM we use today relies on the Transformer. This is the foundation of our work.

[11] Razorpay Dev Team, "Integrating payment gateways in the fintech market " Payments Tech Review, vol. 8 No. 2 Pp. 204–215, 2023. Payments in India have a lot of edge cases. This paper by Razorpay Dev Team was good to read because it showed me how professionals handle security.

[12] K. D. Hunter, For the Win: The psychology of game thinking in business. Wharton Digital Press,Philadelphia 2012. I never thought about combining games with business until I read this book by K. D. Hunter. It is a perspective.

[13] Y. K. Chou, Actionable Gamification: A guide to the Octalysis framework. Octalysis Media, 2015. The Octalysis framework is a way to look at user engagement. This book by Y. K. Chou is helpful for understanding how to engage users.

[14] N. Eyal, Hooked: A guide to building habit- forming products. Portfolio/Penguin, 2014. This book by N. Eyal explains why we can't stop checking apps. It is interesting to learn about habit-forming products.

[15] J. Hamari, "Understanding Gamification: A literature review of studies " in Proc. 47Th Hawaii Int. Conf. System Sciences (HICSS) 2014. This paper by J. Hamari is a look at whether gamification works. It is good to see the research behind gamification.

[16] B. J. Fogg, "The Fogg Behavior Model: A way to understand design " in Proc. 4Th Int. Conf. Persuasive Technology, 2009. Fogg's model is simple. Explains human actions online. This paper by B. J. Fogg is helpful for understanding user behavior.

[17] L. Chen and F. Wang "How AI-based recommender systems are evolving," IEEE Access Journal, vol. 8 Pp. 72064–72084 2020. Recommender systems are getting good at predicting what we want. This paper by L. Chen and F. Wang shows how AI- based recommender systems are evolving.

[18] S. Grider, "Handling streams in web architectures " Web Performance Summit (WPS) San Francisco, 2022. Grider's courses are top-notch for understanding streaming. This talk by S. Grider was really helpful.

[19] SQL-Alchemy Core, "Relational Data Management with ORM " Relational Database Systems Guide, 2023. Using an ORM makes database work cleaner. Sqlalchemy is powerful for Python. This guide by SQL-Alchemy Core is helpful for learning about data management.

[20] P. M. Gollwitzer "Why simple plans lead to results: A study, on implementation intentions " American Psychologist, vol. 54. 493–503, 1999.Simple "if-then" plans can boost productivity. This study, by P. M. Gollwitzer shows how simple plans can lead to results.