Bilkent University
Department of Computer Science

CS491 - Senior Design Project I

Project Specification Document

Fennec Radio
T2439

Süleyman Yağız Başaran
Bartu Özyıldırım
Hikmet Kaan Oktay
Feza Emir Çelik
Atakan Keser

Supervisor: Özgür Ulusoy
Innovation Expert: Ali Cengiz Beğen

Nov. 21, 2024

Table Of Contents

1. Introduction

Traditional radio is static, one-way, and lacks personalization. Listeners have limited control over what they hear, being confined to the schedules and content determined by radio stations. For content creators, the barriers to entry are significant, involving high costs and complex broadcasting technology. Aspiring broadcasters often need to apply to radio companies and meet strict-rich requirements, which can be discouraging.

In a world where audiences crave interactivity, customization, and real-time engagement, radio is struggling to keep up with modern user expectations. Therefore, **FennecRadio** enters the party.

There's a need for a platform that allows users to both broadcast and listen, while incorporating popular and advanced technology like AI to create an interactive, personalized radio experience. Whether it's sharing music, discussing news, or just casual chatting, everyone should have the freedom to create and consume content that resonates with them which is falling behind when the traditional radio is considered.

This project seeks to revolutionize the radio experience by making it more accessible, interactive, and personalized for everyone. Additionally, this report contains a brief description of the project, its architecture and technologies, project constraints, ethical issues, standards, functional and non-functional requirements, market research, academic analysis, and mockups.

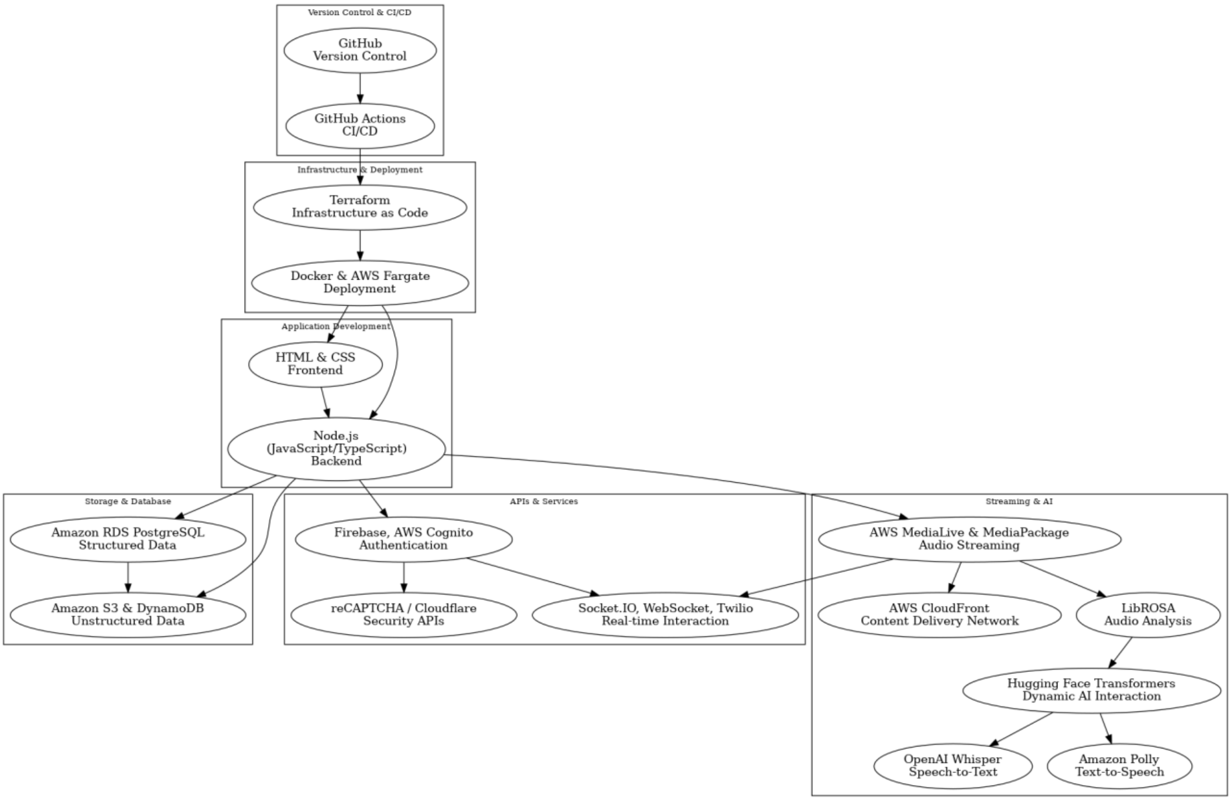
1.1 Description

FennecRadio is an online platform that allows everyone to join live audio broadcasts or start their own shows effortlessly. Whether you want to share your favorite music, talk about the latest news, or have casual chats about everyday topics, FennecRadio gives you the tools to do so. You don't need any special training or experience working with traditional radio stations. This means anyone can become a broadcaster, making the radio experience more inclusive. Furthermore, people can chat with the streamer in real time without waiting for long telephone call delays of the classic radio; even, they can join the broadcast with one button click when streamer decides to get listeners to the broadcast.

The platform also features AI-managed radio channels that provide continuous content, ensuring there's always something interesting to listen to. These AI stations fill in the gaps when live shows aren't available, offering a variety of broadcasts chose for listeners' interests.

Our goal is to **democratize** radio broadcasting by making it accessible and community-driven. We aim to bring new voices to the airwaves and create a space where everyone can participate in radio, regardless of their background. By breaking down traditional broadcasting barriers, FennecRadio hopes to revolutionize how people engage with radio, making it more interactive and diverse for both broadcasters and listeners.

1.2 High Level System Architecture & Components of Proposed Solution



The diagram provides a comprehensive overview of the system architecture, highlighting the key components and their interactions. At the core, the Frontend is built using HTML and CSS to provide the user interface, which communicates with the Backend, implemented in Node.js (JavaScript/TypeScript). The backend serves as the central processing unit, managing interactions with various services. It handles Live Streaming Services, using AWS MediaLive and MediaPackage for audio streaming, complemented by AWS CloudFront for global content delivery. For data storage, the architecture employs Amazon RDS PostgreSQL for structured data and Amazon S3/DynamoDB for unstructured data like files and logs.

The system integrates multiple APIs to enhance functionality. Authentication is managed by Firebase and AWS Cognito, while Socket.IO, WebSocket, and Twilio APIs facilitate real-time interactions such as live chat and notifications. Security is bolstered by reCAPTCHA or Cloudflare. Advanced capabilities are powered by AI, leveraging LibROSA for audio analysis, Hugging Face Transformers for dynamic interactions, OpenAI Whisper for speech-to-text, and Amazon Polly for text-to-speech. This interconnected structure ensures a robust, scalable, and feature-rich platform, tailored for seamless live streaming, secure user interaction, and real-time data processing.

1.3 Constraints

Constraints refer to the specific limitations and conditions that influence the development and operation of the project. These constraints can be categorized into several key areas, each with its own unique set of requirements and considerations. In this section, a detailed overview of these constraints will be covered under Implementation Constraints, Economic Constraints, Ethical Constraints, Maintainability Constraints, Privacy and Safety Constraints.

1.3.1 Implementation Constraints

- Use GitHub for version control to enable collaborative development and code management.
- Use Jira and Slack for task management.
- Implement GitHub Actions for CI/CD to automate testing and deployment processes.
- Define and manage infrastructure using Terraform for consistency and reproducibility.
- Deploy applications using containerization with Docker and orchestration with AWS Fargate.
- For the front-end web application, HTML and CSS will be used. The reason for this choice is that these languages/technologies are the easiest ones to learn and start applying since it is important to start developing the fundamental parts of the project. It is also useful since every member of the project group will help in the front-end parts and making everyone learn a new and complicated front-end language/framework would cost more time.
- The technology for the backend will be Node.js framework implemented with Javascript/Typescript because real-time features are critical as it shines for applications requiring real-time capabilities (e.g., chat, live updates, WebSockets) due to its non-blocking, event-driven architecture. Handles high levels of simultaneous requests efficiently, making it great for scaling with many users. Which is important for future enterprise growth, as Netflix, Uber, LinkedIn etc. also use it. We also considered using Python’s Flask but it is not performing well with high frequency requests and is considered weak for scaling the project.
- For authentication APIs, Firebase Authentication and Twilio will be used.
- For real-time interaction, Socket.IO and Twilio APIs will be used.
- For security APIs reCAPTCHA or Cloudflare can be used.
- Implement user authentication using AWS Cognito with OAuth 2.0 protocols.
- Use AWS Elemental MediaLive and MediaPackage for encoding and delivering live audio streams.
- Utilize AWS CloudFront as a Content Delivery Network (CDN) to ensure low-latency global content delivery.
- Implement WebSocket APIs for real-time features like live chat and instant notifications.
- Store user information and structured data in Amazon RDS PostgreSQL.

- Store images and recorded audio files in Amazon S3 buckets.
- Store chat messages and unstructured data in Amazon DynamoDB.
- Using LibROSA, audio features such as rhythm, tempo, and genre are extracted directly from the tracks.
- By leveraging Hugging Face Transformers, these AI broadcasters can dynamically change the stream. Pre-trained language models such as GPT and BERT will be used.
- To facilitate real-time interactions with AI broadcasters, the platform uses OpenAI Whisper for speech-to-text conversion.
- Amazon Polly is used to synthesize natural-sounding AI broadcaster voices.

1.3.2. Economic Constraints

- Optimize AWS resource usage to stay within the allocated budget constraints.
- Training and deploying machine learning models will be the biggest cost of this project; however, an exact cost could not be calculated due to not knowing how much training is required and potential cost reductions with brand deals and sponsorships.
- Obtaining a domain will have a cost.
- Other services and technologies like Github, Node.js, Figma, Docker etc. are free and some are open source.
- API's are free until a given amount of free calling, however it is uncertain for now because of the unpredictable requirements of the project.

1.3.3 Ethical Constraints

- The data obtained by processing the broadcasts and users' information are confidential and won't be shared by third parties.
- All the data will be stored in a secure place to protect user privacy.
- Temporary storage of streams for a minimum period necessary to review and analyze reported content.
- A long lasting storage for users chat logs will be stored to use in the case of a report or harmful action.
- Establish clear guidelines prohibiting unauthorized sharing of copyrighted content and implement mechanisms to detect and prevent such activities.
- Regularly audit AI models for biases and ensure diversity in training data to promote fairness and equality.
- The Ethics of the National Society of Professional Engineers will be followed by the team in the project [3].

1.4 Professional and Ethical Issues

All users will be informed on which data is collected, and their informed consent will be held in the registration. Transparency will be maintained in how user data is collected, stored, and used within the application. Also hashes will be used to store the passwords to increase privacy and security.

- Setting clear community guidelines and providing tools for reporting violations is crucial.

Not storing stream recordings long-term can hinder your ability to review and address reports of inappropriate or harmful content. This could allow harmful behaviors to go unchecked, negatively impacting users. As a platform provider, we have an ethical obligation to ensure that content shared on FennecRadio does not include hate speech, harassment, illegal activities, or other harmful material and will be checked if further investigation is needed in the case of a report.

- Avoid algorithmic biases and provide an equal streaming platform.

We will follow BTK, RTÜK and KVKK protocols to ensure everyone is safe, secure, private and also as free as possible.

1.5 Standards

Throughout the project, IEEE 754 Standards will be present like maintaining floating-point consistency across components. We'll use Git to manage all our source code and related files, with the repositories hosted on GitHub. They will follow semantic versioning to track the clarity of updates made to the versions. The team will utilize two-week long Scrum points including stand ups and backlog sprints reviews, and hold Sprint review retrospective and planning meetings at the end of each Sprint retrospectives. SHA-256 Encryption, every sensitive user data will be hashed and salted with.

Lastly, and one of the most important aspects, we will follow the rules of RTÜK about legal and streaming protocols.

2. Design Requirements

2.1 Functional Requirements

- The user can register/login using their email or using cognito.
- The user can reset their password using their registered email.
- The user can verify their email address to activate their account.
- The user can access and edit their personal profile information, including username, profile picture, and bio.

- The user can browse and listen to live radio broadcasts from various channels.
- The user can search for broadcasts by topic, genre, keyword, or broadcaster name.
- The user can see a list of featured, popular, or recommended broadcasts based on their interests, including AI powered disk-jockeys.
- The user can adjust audio settings such as volume and streaming quality.
- The user can participate in live chat during broadcasts.
- The user can ask streamers to join the broadcast live.
- The user can follow broadcasters or channels to receive updates and notifications.
- The user can create their own radio channel for broadcasting.
- The user can set up and schedule live broadcasts.
- The user can stream live audio content using a microphone
- The user can manage their broadcasts, including editing titles, banners, descriptions, and tags.
- The user can upload images for their channel logo or broadcast cover art.
- The user can start events such as polls, games etc. in order to make the stream more interactive.
- The user can access help resources or contact customer support.
- The user can report a radio channel or another user for inappropriate behavior.

2.2 Non-Functional Requirements

2.2.1 Usability

FennecRadio is designed to be simple and user-friendly; the platform promises an intuitive and enjoyable experience for all users. It is really important because we are aware of the fact that FennecRadio will be thriving in the entertainment sector and it must be designed according to that. Users can easily sign up, navigate in the platform, and start or join broadcasts without any “user’s guide”. In just 3 clicks they can start their stream or start watching their favorite streamer. The interface should be intuitive and work smoothly across all devices meaning, the platform should be responsive, functioning seamlessly on desktop, tablet, mobile and car-players to cater to a diverse audience.

2.2.2 Reliability

The system must maintain high availability, ensuring minimal downtime for all services because people can start their live streams or join others' live streams whenever and wherever they want 24/7. Real-time streaming and interactions must be fault-tolerant to handle unexpected disruptions smoothly. Streaming services must provide smooth and consistent playback with minimal buffering, even if a user is having a hard time under varying network conditions. Regular backups and automated recovery processes must be in place to prevent data loss. Users should be notified clearly in the event of errors, with an option to report the issue for resolution.

2.2.3 Performance

Performance is a critical aspect of FennecRadio. The application must ensure consistent accessibility and responsiveness for users across all geographical locations, providing the same high-quality experience regardless of where they are. The system must provide fast response times for user actions, such as loading playlists, starting streams, or interacting with live features, with minimal latency. By leveraging global Content Delivery Networks (CDNs) like AWS CloudFront, we can distribute content efficiently and minimize latency, ensuring that streaming and interactive features operate at the same work rate worldwide.. Streaming must deliver consistent audio quality, even during peak usage, by leveraging AWS Elemental MediaLive for its low-latency and high-reliability streaming capabilities. Additionally, DynamoDB is integrated to support high-speed retrieval of analytics data for real-time updates without performance degradation. The decision to use Node.js instead of Flask for the backend occurs from their lightweight, scalable architectures that excel at handling concurrent user requests efficiently. Node.js, with its event-driven nature, is particularly suited for real-time interactions like live chat or notifications.

2.2.4 Supportability

The platform will also be accessible by mobile and desktop devices so as to accommodate a greater number of users but our main concern is web at first sight. Modular design principles will be followed to ensure easy integration of future platforms, such as in-car entertainment systems. The system will use a modular design to allow for easy updates and additions of new features without disrupting existing functionality. All components of the platform will be thoroughly documented, including code comments, developer guides, and user manuals, to assist in future maintenance, improvements and support. The platform will be built to handle growth in user numbers and broadcast content, enabling seamless scaling as demand increases. Code will adhere to industry best practices and standards to ensure it is clean, well-organized, and easy for developers to understand and modify. Testing APIs and Object Oriented Programming like paradigms will be followed to make the codebase easier to understand and solve if a problem occurs. Lastly, regular data backups and recovery plans will be in place to prevent data loss and ensure the platform can quickly recover from any failures.

2.2.5. Scalability

Scalability at its core to accommodate a growing user base and increasing content without compromising performance. By leveraging cloud-based infrastructure and services like AWS Fargate and DynamoDB, the platform can automatically scale computing resources in response to demand. The application's microservices (e.g. RabbitMQ) architecture allows individual components to scale independently, optimizing resource utilization and maintaining its performance worldwide. This scalable design ensures that FennecRadio can handle spikes in traffic and continue to deliver high-quality streaming and interactive features as it expands. To achieve those, we chose Node.js over Flask for our backend development. It utilizes a non-blocking, event-driven architecture, which makes it highly effective for managing multiple simultaneous connections—a common requirement for live audio streaming platforms like ours. This design allows Node.js to handle real-time data and scale horizontally with ease ensuring that FennecRadio can seamlessly scale to accommodate increasing user demand, providing a smooth and responsive experience for both broadcasters and listeners..

2.2.6 Security & Privacy

FennecRadio will store user data to provide a personalized and interactive radio experience, including names, interests, passwords, emails, listening history, chat messages, and other personal information. It is crucial that our systems are highly secure to protect this sensitive data and that we handle such information with the utmost care. To achieve this, we will employ advanced encryption algorithms to secure data both at rest and in transit, ensuring that all user information is protected from unauthorized access. We will implement secure authentication mechanisms, such as OAuth 2.0 via AWS Cognito, to verify user identities and prevent unauthorized account access. This includes measures to prevent CSRF (Cross-Site Request Forgery) attacks, ensuring that unauthorized commands cannot be transmitted through our platform. Additionally, we will establish rate limiting and monitoring mechanisms to protect our servers from excessive requests and potential denial-of-service attacks. We are committed to complying with all relevant data protection regulations, such as GDPR and CCPA, and will seek explicit user consent whenever necessary to balance personalization and privacy. By integrating these robust security measures—including data encryption and secure authentication—FennecRadio aims to build trust with our users and provide a safe and secure environment for everyone.

3. Feasibility Discussions

3.1 Market & Competitive Analysis

3.1.1 Market Analysis

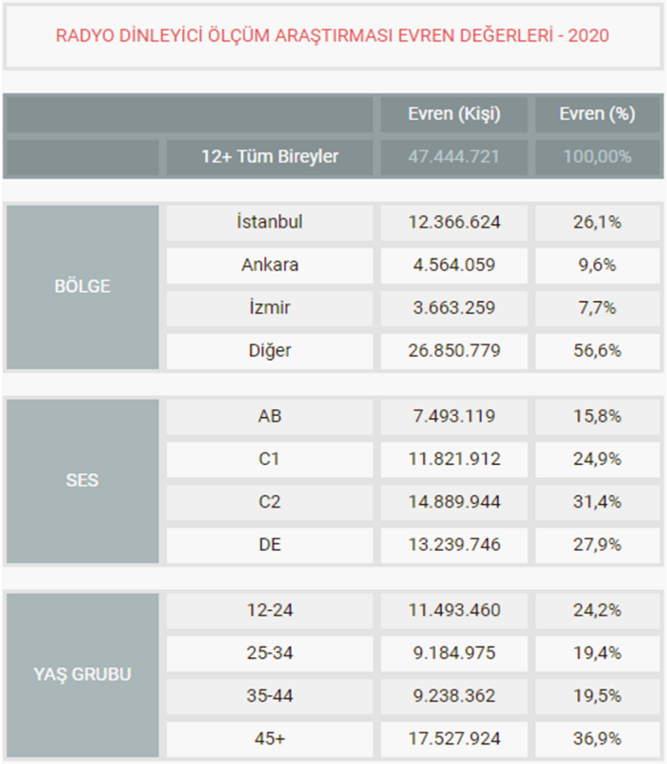
Despite the decline in traditional radio popularity, it remains a significant medium globally. In the U.S., more Americans listen to the radio weekly than use Facebook, with 55% of Gen Z listening to AM/FM radio daily. Adults listen 104 minutes of radio per day and spend an average of 12.2 hours per week listening to the radio, supported by the growth in smart speakers and online radio, which is also a huge number as 100 million Americans own a smart speaker [4].

Radio holds the highest share of collective trust across all advertising channels. This highlights radio's sustained relevance, particularly with trusted advertising, as 77% of listeners act on endorsements by their favorite radio personalities [4].

By merging traditional radio's strengths with interactive digital features, Fennec Radio is well-placed to tap into both older and younger generations. Smart speakers and streaming technologies offer new avenues to reach tech-savvy audiences while maintaining appeal to existing radio users.

Key statistics supporting market potential:

- Radio advertising is a $36.1 billion global market, offering monetization opportunities for our freemium and ad-supported model.
- The Turkish market presents a significant opportunity, as radio remains widely used across generations.



RADYO DİNLEYİCİ ÖLÇÜM ARAŞTIRMASI EVREN DEĞERLERİ - 2020

		Evren (Kişi)	Evren (%)
	12+ Tüm Bireyler	47.444.721	100,00%
BÖLGE	İstanbul	12.366.624	26,1%
	Ankara	4.564.059	9,6%
	İzmir	3.663.259	7,7%
	Diğer	26.850.779	56,6%
SES	AB	7.493.119	15,8%
	C1	11.821.912	24,9%
	C2	14.889.944	31,4%
	DE	13.239.746	27,9%
YAŞ GRUBU	12-24	11.493.460	24,2%
	25-34	9.184.975	19,4%
	35-44	9.238.362	19,5%
	45+	17.527.924	36,9%

Figure 1: 2020 Turkiye Radio Listener Map [5]

3.1.1 Competitive Analysis

Fennec Radio's competition landscape is diverse, ranging from global live-streaming giants to niche, regional platforms. This section analyzes these competitors to highlight Fennec's market positioning and opportunities.

3.1.1.1 Primary Competitors: Global Live Broadcasting Giants

Fennec Radio's main competition comes from established live-streaming platforms that dominate the market, though their focus is primarily on video content rather than interactive radio experiences.

Twitch [7]:

- 2024 Statistics:
  - Average Concurrent Viewers: **2.38 million**
  - Concurrent Channels: **96,300**
  - Monthly Broadcasters: **7.44 million**
- Turkish Market Presence:
  - The Turkish audience on Twitch is significant, with channels like "Jahrein" achieving an average of **99,459 viewers [9]**.
  - The top 10 Turkish Twitch channels account for **25% of the total Turkish viewership**, highlighting the platform's strong local impact [8].
- Opportunity for Fennec: Twitch's video-centric model leaves room for audio-focused, interactive platforms like Fennec Radio to fill the gap in personalized and engaging radio broadcasting.

YouTube Live:

- Türkiye Statistics [10]:
  - Peak concurrent viewers for popular live events include (October 2024):
    - Star TV: **82,672**
    - Show TV: **48,412**
    - ATV: **42,391**
  - Weekly live stream popularity in Türkiye showcases the demand for live content but lacks a focus on interactive, audio-based engagement.
- Opportunity for Fennec: YouTube Live’s lack of dedicated radio-style interaction allows Fennec to capture audiences seeking personalized live experiences.

Kick [9]:

- An emerging platform competing with Twitch, known for better revenue sharing for creators.
- Opportunity for Fennec: Kick is primarily video-focused, and Fennec can stand out by focusing on a unique radio-like broadcasting experience that leverages audio interaction and AI-driven features.

3.1.1.2 Secondary Competitors: Regional and Niche Platforms

Smaller platforms cater to specific audiences or regions, offering a variety of features but often lack global scalability. Fennec can start from locally like the below platforms and then scale into the global market.

Platform	Viewers	Channels	Market Focus
NimoTV	60,569	300	84% Vietnamese audience
Trovo	8,457	497	83% Russian audience
Rumble	25,507	214	Mostly English-speaking
Bigo LIVE	1,345	91	Mostly English-speaking
AfreecaTV	48,529	1,080	100% Korean audience
Naver's CHZZK	125,925	4,041	100% Korean audience
SOOP	3,808	50	92% Thai audience
SteamTV	34,765	213	100% English-speaking

Table 1: Monthly statistics of the platforms [6]

These platforms focus heavily on video-based live streaming for niche markets. Their localized appeal limits their expansion to other demographics or regions.

- Opportunity for Fennec: Fennec Radio's global scalability and focus on audio interaction and AI-powered customization allow it to tap into these niche markets while differentiating itself with an innovative radio-style experience.

This table means there's local demand for streaming services all around the globe, which Fennec can take advantage of.

3.1.1.3 Competitive Advantage of Fennec Radio

Differentiation from Competitors:

- Personalization: Fennec's AI-powered DJs and personalized playlists create a unique user experience unmatched by existing video-based platforms.
- Accessibility: By removing technical barriers (no third-party software like OBS required), Fennec democratizes broadcasting for everyone.
- Interactivity: Real-time features like live chats, polls, and audience participation foster a stronger connection between broadcasters and listeners, going beyond passive consumption.
- Radio Legacy: The "radio" branding taps into the nostalgia of older generations while modernizing it for Gen Z and millennials with advanced technologies like streaming.

3.2 Academic Analysis

The Fennec Radio platform is built on a strong academic foundation, utilizing state-of-the-art technologies and methodologies that have been extensively validated in research. These approaches ensure that the platform is not only innovative but also reliable, scalable, and user-focused.

The personalization engine at the core of Fennec Radio combines collaborative filtering and content-based filtering. These techniques are widely recognized for their efficiency in creating tailored recommendations. Hybrid systems that blend these approaches have been shown to solve common challenges like the cold start problem while enhancing user satisfaction [11]. Additionally, deep learning methods such as Convolutional Neural Networks (CNNs) are employed to analyze audio data. These models excel at capturing complex audio features, as demonstrated in research on music classification [12]. This combination ensures that Fennec Radio provides users with highly relevant and engaging content.

To enhance user interaction, the platform integrates cutting-edge artificial intelligence. Pre-trained transformer models like GPT and BERT are used to power AI broadcasters, enabling them to engage in meaningful, natural conversations with users. These models, considered state-of-the-art in natural language processing, have been validated for their ability to understand and generate human-like language [13]. This makes Fennec Radio’s AI broadcasters not only functional but also interactive and responsive, fostering deeper engagement.

The platform also incorporates real-time speech recognition and text-to-speech synthesis, enabling seamless voice interactions. Technologies like OpenAI’s Whisper and Amazon Polly are based on methodologies proven in academic research. For instance, end-to-end speech recognition models have been shown to deliver accurate voice command processing [14], while neural network-based text-to-speech systems produce natural, human-like audio outputs that enhance user satisfaction.

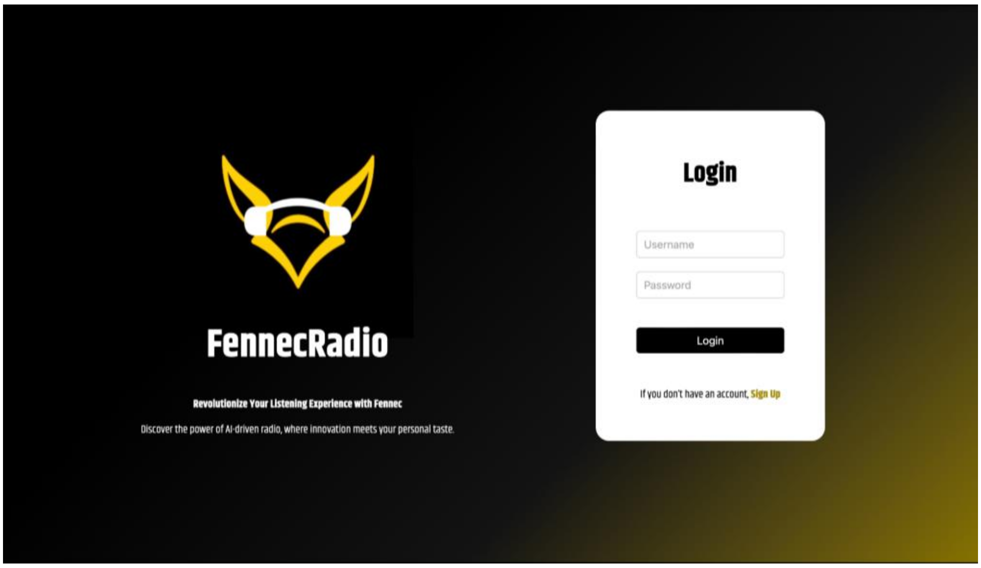
To handle large-scale data streaming, Fennec Radio relies on Apache Kafka, a robust framework for handling real-time data streams. Kafka’s publish-subscribe model is a staple in distributed computing, known for its ability to process large-scale data with reliability and low latency [15]. This makes it an ideal choice for managing real-time interactions and content delivery on the platform, ensuring responsiveness even as user demand grows.

The user-centric design of Fennec Radio is guided by principles from Human-Computer Interaction (HCI) research. Usability studies have shown that simplifying interfaces and minimizing the learning curve are critical for broad user adoption [16]. In line with these insights, Fennec Radio’s interface is designed to be intuitive and accessible, enabling users of all technical backgrounds to engage with the platform effectively.

Ethical considerations are also a key component of Fennec Radio. By adhering to GDPR standards and implementing privacy safeguards, the platform aligns with academic recommendations on ethical AI use [17]. Transparency, accountability, and user privacy are prioritized to build trust and ensure responsible application of AI technologies.

This robust academic foundation underscores the feasibility and effectiveness of the Fennec Radio platform. By leveraging proven methodologies in recommendation systems, AI interactions, real-time processing, and user design, the platform delivers a seamless and engaging experience. Moreover, its commitment to ethical practices ensures that Fennec Radio is not only innovative but also responsible in its approach to modern technology.

4. Mockups

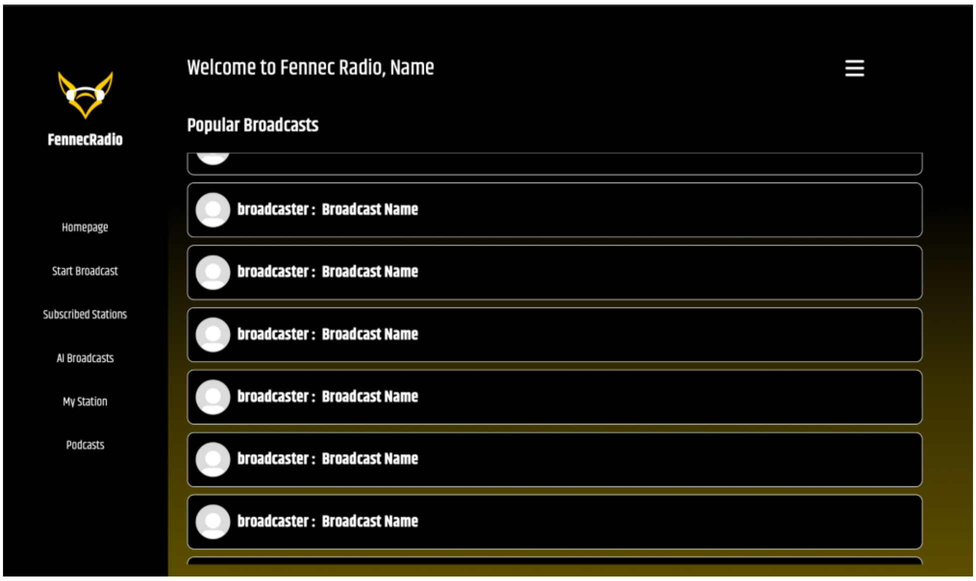


Revolutionize Your Listening Experience with Fennec

Discover the power of AI-driven radio, where innovation meets your personal taste.

If you don't have an account, Sign Up

This is the login page designed and tested, tested and working.



Welcome to Fennec Radio, Name

Popular Broadcasts

broadcaster : Broadcast Name

broadcaster : Broadcast Name

broadcaster : Broadcast Name

broadcaster : Broadcast Name

broadcaster : Broadcast Name

broadcaster : Broadcast Name

This is the planned front-page when user logins. User can see recommended streamers and watch them or choose other actions from the bar left. Tested and working.

5. Glossary

AWS: Amazon Web Services

6. References

[1] AWS. "Amazon IVS: Build Interactive Video Experiences." Accessed November 22, 2024. https://aws.amazon.com/ivs/.

[2] TrackIt. "Amazon IVS or AWS Elemental MediaLive for Streaming?" Accessed November 22, 2024. https://trackit.io/amazon-ivs-or-aws-elemental-medialive-streaming/.

[3]NSPE, “NSPE Code of Ethics for Engineers,” National Society of Professional Engineers, Jul. 2019. accessed Nov. 22, 2024. https://www.nspe.org/resources/ethics/code-ethics.

[4] Musical Pursuits. "Radio Statistics and Trends." Accessed November 22, 2024. https://musicalpursuits.com/radio/.

[5] Riak. "Evren Değerleri." Accessed November 22, 2024. https://riak.com.tr/evren-degerleri.

[6] Streams Charts. "Platform Statistics." Accessed November 22, 2024. https://streamscharts.com/platforms.

[7] Twitch Tracker. "Twitch Statistics." Accessed November 22, 2024. https://twitchtracker.com/statistics.

[8] Twitch Tracker. "Twitch Turkish Language Statistics." Accessed November 22, 2024. https://twitchtracker.com/languages/Turkish.

[9] Twitch Metrics. "Kick Channels Popularity." Accessed November 22, 2024. https://www.twitchmetrics.net/kick_channels/popularity?lang=Turkish.

[10] Playboard. "Most Watched Live Videos in Turkey - Weekly." Accessed November 22, 2024. https://playboard.co/en/chart/live/most-watched-all-videos-in-turkey-weekly.

[11] Burke, R. (2002). Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction, 12(4), 331-370.

[12] Choi, K., Fazekas, G., Sandler, M., & Cho, K. (2017). Convolutional Recurrent Neural Networks for Music Classification. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

[13] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All You Need. Advances in Neural Information Processing Systems (NeurIPS).

[14] Graves, A., Mohamed, A., & Hinton, G. (2013). Speech Recognition with Deep Recurrent Neural Networks. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).

[15] Kreps, J., Narkhede, N., & Rao, J. (2011). Kafka: A Distributed Messaging System for Log Processing. Proceedings of the NetDB Conference.

[16] Nielsen, J. (1993). Usability Engineering. Morgan Kaufmann Publishers.

[17] Floridi, L., Cowls, J., Beltrametti, M., Chatila, R., Chazerand, P., Dignum, V., Luetge, C., Madelin, R., Pagallo, U., Rossi, F., Schafer, B., Valcke, P., & Vayena, E. (2018).

[18] AI4People—An Ethical Framework for a Good AI Society: Opportunities, Risks, Principles, and Recommendations. Minds and Machines, 28(4), 689-707.