



An-Najah National University
Faculty of Engineering And Information Technology
Department of Computer Engineering

**Presented in partial fulfillment of the requirements for
Bachelor degree in Computer Engineering**

Athena

**Tala AbuSoud
Raya Thawabe**

Supervisor DR. Samer Arandi

June 2025

Acknowledgements

We would like to express our heartfelt gratitude to all who supported us throughout the completion of this project.

We express deepest thanks to our supervisor, Dr. Samer Arandi, whose guidance, encouragement, and invaluable insights were instrumental in the success of this work. His expertise and mentorship have been a constant source of motivation and direction.

We are also profoundly grateful to our families and friends for their unwavering support, patience, and encouragement throughout this journey. Their belief in us has been the cornerstone of our perseverance and achievement.

Disclaimer Statment

This report was written by Tala AbuSoud & Raya Thawabe at the Computer Engineering Department, Faculty of Engineering, An-Najah National University. It has not been altered or corrected, other than editorial corrections, as a result of assessment and it may contain language as well as content errors. The views expressed in it together with any outcomes and recommendations are solely those of the students. An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was commissioned.

List of Figures

3.1	Athena System Architecture	6
3.2	ERD of Athena DB	9
3.3	ERD of dejavu DB	10
3.4	Splash Screen	10
3.5	Sign In Page	10
3.6	Sign Up Page	10
3.7	Password doesn't match Message	11
3.8	Invalid Phone Number Message	11
3.9	Email Verification Popup	11
3.10	Forget Password	12
3.11	Reset code	12
3.12	Reset Password	12
3.13	User Log-in app successfully	12
3.14	Complete profile: Birth date & Gender	12
3.15	Complete profile: Profile Picture	12
3.16	Complete profile: User interests 1	13
3.17	Complete profile: User interests 2	13
3.18	Home Page 1	13
3.19	Home Page 2	13
3.20	All Latest Episodes	14
3.21	All Recently Played	14
3.22	User Profile page	14
3.23	User edits profile	14
3.24	User's Notifications	14
3.25	The form to convert to a channel account	15
3.26	Pending Request to Convert to Channel	15
3.27	Library page	15
3.28	All liked episodes	15
3.29	All saved podcasts	15
3.30	Recommendations (Before Select Mood)	16
3.31	Mood Tracker – User Select the Current Mood	16
3.32	Updated Recommendations Based on Mood	16
3.33	All Regular Channels	17
3.34	All External Channels	17
3.35	Channel Page With Details	18
3.36	All Podcasts in Channel	18
3.37	Recommendations (Before Select Mood)	19
3.38	Mood Tracker – User Select the Current Mood	19
3.39	Updated Recommendations Based on Mood	19
3.40	External Episode 1	20
3.41	External Episode 2	20

3.42	Listen External Episode	20
3.43	Regular Episode 1	20
3.44	Regular Episode 2	20
3.45	Bad Review Rejected	21
3.46	Good Review Submitted	21
3.47	Listen Regular Episode	22
3.48	Audio Playback Speed	22
3.49	Set Sleep Timer	22
3.50	Share Episode	22
3.51	Episode Transcript	23
3.52	Chatbot 1	23
3.53	Chatbot 2	23
3.54	Chatbot 3	23
3.55	Search & All Categories	24
3.56	Category Page	24
3.57	Searching	24
3.58	Record Short Audio Clip	25
3.59	The Result Of Matching	25
3.60	Channel Profile 1	26
3.61	Channel Profile 2	26
3.62	Channel Notifications	26
3.63	All Podcasts	26
3.64	Create New Podcast	26
3.65	Podcast Created	26
3.66	Delete Podcast	27
3.67	Podcast Page	27
3.68	Edit Podcast	27
3.69	Create New Episode 1	28
3.70	Create New Episode 2	28
3.71	Wait Message	28
3.72	Episode Page.1	28
3.73	Episode Page.2	28
3.74	Listen Episode	28
3.75	Edit Episode	29
3.76	Delete Episode	29
3.77	All External Channels	30
3.78	Create External Channel	30
3.79	External Channel Page	30
3.80	Create External Podcast 1	30
3.81	Create External Podcast 2	30
3.82	External Podcast Page	30
3.83	Create External Episode 1	31
3.84	Create External Episode 2	31

3.85	External Episode Page	31
3.86	Admin Profile	32
3.87	Login Page	32
3.88	All Users Table	33
3.89	User Details Part.1	33
3.90	User Details Part.2	34
3.91	All Requested Channels Part.1	34
3.92	All Requested Channels Part.2	35
3.93	All Channels	35
3.94	All Pending Channels	36
3.95	History Part.1	36
3.96	History Part.2	37
3.97	Channel Details	37
3.98	All Podcasts	38
3.99	Podcast Details	38
3.100	All Episodes	39
3.101	Episode Details	39
3.102	Episode Speakers and Script	40
3.103	Insights Part.1	41
3.104	Insights Part.2	41
3.105	Insights Part.3	42
3.106	Insights Part.4	42
3.107	Insights Part.5	43
3.108	Ads Manager Part.1	44
3.109	Ads Manager Part.2	44
3.110	All Campaigns - Ads Manager Part.3	45
3.111	Edit Campaign - Ads Manager Part.4	45
3.112	Renew Campaign - Ads Manager Part.5	46
3.113	Download All Campaigns Data as a CSV - Ads Manager Part.6	46
3.114	Example of CSV - Ads Manager Part.7	46

List of Tables

Contents

Acknowledgements	ii
Disclaimer	iii
List of Figures	vi
List of Tables	vi
Abstract	viii
1 Introduction	1
1.1 Background	1
1.2 Objectives	1
1.3 Significance	2
1.4 Organization of the Report	3
2 Theoretical Background and Previous Work	4
2.1 Theoretical Background	4
2.2 Previous Work	4
2.2.1 Global Podcast Platforms	4
2.2.2 Middle East Platforms	4
3 Methodology	6
3.1 System Architecture and Design	6
3.1.1 System Architecture	6
3.1.2 System Design	7
3.2 Technologies and Programming Languages	7
3.2.1 Front-end	7
3.2.2 Back-end and Tools	8
3.3 Database	9
3.4 Features Implementation	10
3.4.1 Mobile Application	10
3.4.2 Web Dashboard	32
3.5 Standards and Specifications	47
3.6 Constraints	47
3.6.1 Economy	47
3.6.2 Environment	47
3.6.3 Society	47
3.6.4 Ethics and Privacy	47
4 Results	48

5	Discussion	49
6	Conclusions & Recommendation	50
6.1	Conclusions	50
6.2	Recommendation	50
	References	51

Abstract

The podcast is among the strongest platforms for education, entertainment, and information. With this aim, Athena has been developed as a niche audio content platform to empower independent creators and provide a richer listening experience for the audience. Unlike the mainstream platforms, Athena caters to different user needs with a blend of localized and niche-specific features.

Athena was developed via an iterative life cycle model approach, planning, and initial research to understand user and market needs, followed by UI/UX design, back-end and front-end development, and appropriate testing. The system implements Flutter in the mobile and web front-end, Node.js and PostgreSQL in the back-end, and two Python-based FastAPI microservices for content recommendation and audio fingerprint matching.

Further with real-time transcription supported by OpenAI Whisper and recommendation engine powered by TF-IDF and cosine similarity, audio fingerprinting is done by Dejavu, while a GPT chatbot supports summary, translation, and QA functionalities. OpenAI tools moderate user reviews; Cloudinary handles media streaming and caching.

Two FastAPI-based microservices were developed: a recommender system that uses user interactions, text similarity, and mood data, and a matcher system that uses fingerprinting to identify podcasts from short audio clips. Additional features include notifications, JWT-secured authentication, an admin dashboard to explore app and user insights, manage ads, and channel approvals, and a YouTube

content importer to make external channels content and also converts video links into playable audio episodes.

The final system achieved high performance across all modules: a transcription word error rate of 4.3%, accurate fingerprint matching for clips over 6 seconds, and a sub-300ms average response time for recommendations. The audio matching system delivered accurate identification for clips longer than six seconds with minimal noise. Notifications were triggered based on real-time events and maintained using scheduled cleanup processes. Admins could export ad campaigns as CSV reports.

Athena goes beyond platforms like Spotify and Apple Podcasts by focusing on solo creators, community features, and smart discovery. It delivers a scalable, functional system with future potential for collaborative filtering, monetization, voice interfaces, and deep learning—showcasing the promise of AI-driven podcasting.

1 Introduction

1.1 Background

The podcast industry has grown quickly, becoming a major platform for self-improvement, entertainment, and education. Users who use podcasts to broaden their knowledge, keep informed, or enhance their mindset, often while multitasking, are increasing as on-demand audio platforms gain popularity.

According to world studies by *DemandSage, 2025, Podcaststatistics Team, 2025* and *Backlinko, 2025*, in 2025 there will be nearly 584.1 million regular podcast listeners, whose audience appreciates the convenience, variety, and novelty of audio content. In the Middle East region as a whole, recent statistics by *Northwestern University in Qatar, 2022, Podcast Inc, 2024* and *Shaher, 2024* determine that consumption of podcasts is increasing consistently, especially among Arabs and youngsters. Emiratis and Lebanese people have high levels of listening, predicting growing regional market demand.

Also, AI has transformed how we listen to digital content as chatbots, search by audio fingerprint, and intelligent recommendations become standard applications. This change enables podcast systems that are not only practical, but also intelligent and personalized for each listener.

This vision gave birth to Athena. This AI-powered podcast app provides a more advanced and customized listening experience. With searchable audio, real-time transcripts, a chatbot that summaries and translates content, and mood-based recommendations, Athena hopes to make podcasting engaging and the preferred app for audiences in Arabic and English.

1.2 Objectives

The aim is to design and implement **Athena**, an intelligent, user-focused podcast application that blends cross-platform accessibility with cutting-edge AI features. The key objectives include:

- Developing a clean, user-friendly mobile application using Flutter.
- Having a Node.js backend with PostgreSQL to have extensive analytics, content management, and user authentication.
- **AI-enhanced features**, including:
 - A mood-tracking and interest-tracking recommendation system based on TF-IDF.

-
- A mood-tracking recommendation feature that enables recommendations to be customized to a user’s current emotional state.
 - Dejavu-based audio fingerprinting and Shazam-style matching.
 - OpenAI Whisper is used to transcribe podcast scripts to create interactive real-time transcripts.
 - An AI chatbot to translate, summarize, and answer questions regarding the content.
 - Deploying a smart **ad management system** that:
 - Allows admins to create campaigns, analyze ad performance, and inject ads during playback.
 - Three user roles are maintained:
 - **Admin:** Oversees campaigns, insights, moderation through web dashboard, and and external content management through mobile application.
 - **Channel Accounts:** upload and manage their own podcasts and episodes (with admin approval for conversion user account to channel account).
 - **Regular Users:**Stream, discover, interact and interact with AI-enhanced content.

Lastly, Athena aims to present a scalable, modern podcast platform that provides a uniquely smart and engaging listening experience.

1.3 Significance

The podcast space is booming, yet the majority of platforms fail to leverage AI to improve content discoverability, engagement, and accessibility. Athena bridges this gap by introducing features that cater to today’s digital needs:

- **Personalization:** Unlike standard platforms like Spotify that recommend content, Athena introduces deeper personalization by integrating user moods, interests, and behavioral history to offer enhanced suggestions.
- **Real-Time Transcripts:** Most platforms lack transcript support. Athena provides word-level, synced transcripts to enable searchable and interactive playback.

-
- **Shazam-like feature:** Users can record short audio clips and directly discover matching episodes via audio fingerprinting.
 - **Language support:** Athena is set up to serve both Arabic and English speakers, filling a huge gap in local podcast availability. However, the platform is fully extensible, adding episodes in any language is straightforward, so Athena can be localized or internationalized depending on audience needs and marketing focus.
 - **Enhanced Content Interaction:** Athena’s AI-based chatbot allows users to engage with episode content in the form of summaries, translations, or questions to clarify.
 - **Intelligent Ad Management System:** Unlike traditional ads, Athena’s smart ad platform injects targeted audio ads and offers analytics for creators and advertisers.
 - **Review Moderation:** Athena automatically filters out harmful or inappropriate reviews using OpenAI’s moderation tools and a custom blacklist, promoting a positive and respectful user environment.

The project is particularly significant in educational and cultural contexts, offering a way for users to learn, explore new topics, and improve their mental health through curated audio — anytime and anywhere.

1.4 Organization of the Report

This report is organized into six top-level chapters. It begins with an introduction presenting the background, goals, and importance of the Athena project. The second chapter discusses applicable technologies and past work on podcast platforms, as well as AI integration. The third chapter outlines the tools, system architecture, and development process employed. Chapter four presents the additional features and significant results. Chapter five provides an overview of the outcomes and difficulties. The last chapter summarizes the project, key contributions, and recommendations for future possible improvements.

2 Theoretical Background and Previous Work

2.1 Theoretical Background

Podcasts are a great medium for distributing audio-based content for learning, entertainment, or news. People prefer using podcasts because they can listen whenever, on the go, while driving, or even sitting in one place. Everybody can create and distribute a podcast today with mobile apps and easy-to-use tools. With so many options out there, people want smart apps that allow them to discover and access the content they want.

Athena uses AI to enhance the podcast experience. For example, it will transcribe voice into text through speech-to-text technology so one can read or search for an episode. It recommends episodes based on what one is in the mood to listen to or wants to hear. If one records a clip, Athena is able to identify the entire episode through audio matching. It also has a chatbot function that enables it to summarize or translate the episode, and ask questions about episode content. Also, it play smart ads based on the listener. All these functions make Athena more intimate and more accessible to all its users.

2.2 Previous Work

2.2.1 Global Podcast Platforms

Many podcast platforms exist today, but they each have limitations:

- **Spotify** has strong recommendation features but doesn't offer interactive scripts, chatbot features, or mood-based filtering.
- **Apple Podcasts** focuses on clean design and subscriptions but doesn't provide AI-driven recommendations or live script interaction.
- **Castbox** includes in-audio search using transcripts but lacks Arabic support, mood features, or smart ads.
- **Shazam** is great for recognizing music using audio fingerprints, but it hasn't been adapted for podcasts.

2.2.2 Middle East Platforms

Podcasting has seen a steep increase in the Middle East during the last couple of years, especially in Saudi Arabia, the UAE, and Lebanon. Most Arabic-speaking listeners, especially the younger

generation, are increasingly involved with listening to content. The majority of local podcasting platforms are, however, basic and offer simple audio streaming without international apps' advanced AI features. In fact, many creators and listeners in the region still use YouTube to share and discover podcast-style content, since it's familiar, easy to access, and widely used across Arab countries.

Athena was created to fill this gap by offering a smarter, more interactive podcast experience for both Arabic and English users. It features word-by-word transcripts, mood-based recommendations, and audio clip matching, which is quite similar to Shazam. Users can also talk to an AI assistant to summarize or translate programs, and discover content related to their tastes and listening patterns. With all these features rolled into one app, Athena is one of the first platforms in the region, indeed the world, to introduce AI-powered podcasting into one mobile experience.

3 Methodology

3.1 System Architecture and Design

3.1.1 System Architecture

Athena is designed using a client-server architecture with a RESTful API.

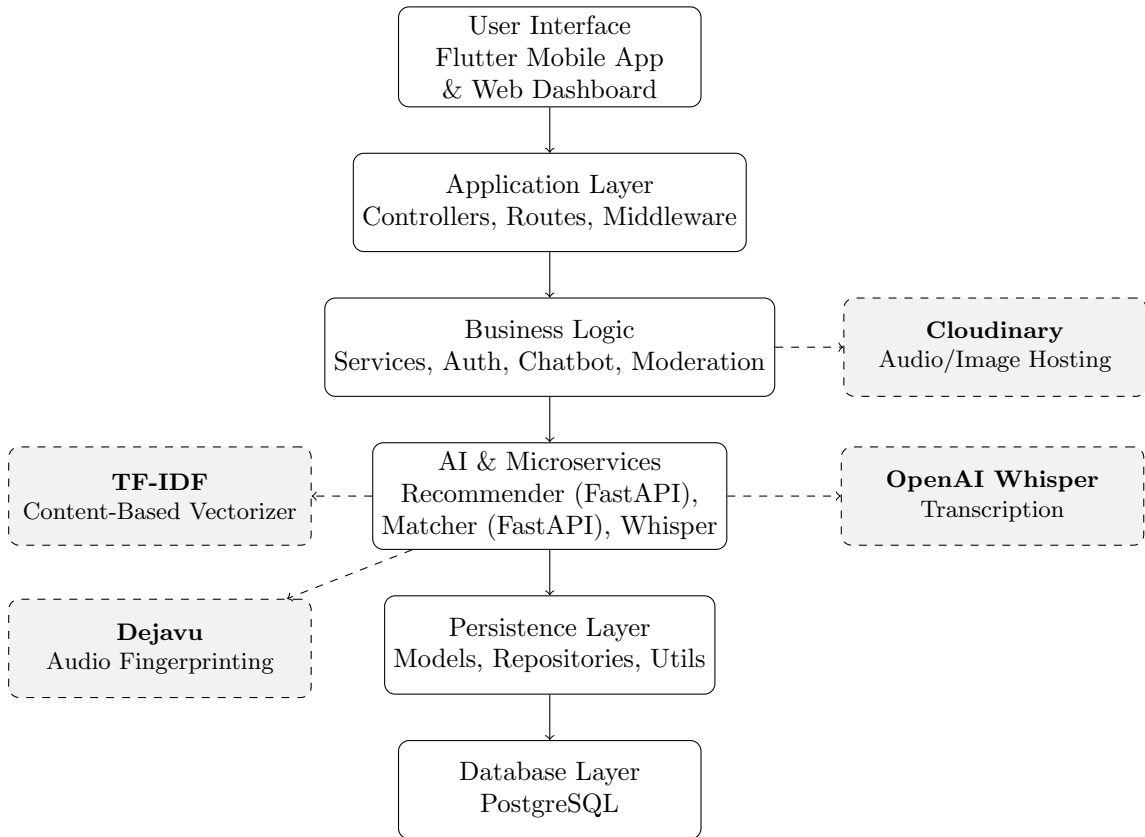


Figure 3.1: Athena System Architecture

The solution comprises three main components:

- **Mobile Front-end:** Developed with Flutter for Android. It encompasses the basic interface through which users will interact with the app.
- **Web Dashboard:** Also built using Flutter, this interface is designed for admin to manage podcasts, episodes, ads, and approvals.
- **Backend Services:** Built with Node.js and Express.js, API request management, authentication, and business logic.

-
- **Database:** PostgreSQL is used to store structured data like user data, episodes, podcasts, and interactions.

All API requests from the Flutter app are managed through Express, which takes care of operations like authentication (JWT), file uploads (Multer), and audio storage/retrieval (Cloudinary). Recommendation microservices and audio matching microservices run standalone using FastAPI.

3.1.2 System Design

The backend is modular, with folders like `controllers`, `models`, `routes`, `middlewares`, and `services`. AI features such as transcription (Whisper), fingerprint matching (Dejavu), and content moderation (OpenAI) are integrated as utilities. A JWT-based system secures endpoints. Admins can manage users, episodes, and ad campaigns through a web interface.

3.2 Technologies and Programming Languages

3.2.1 Front-end

- **Flutter (Dart):** Cross-platform mobile UI.
- **just_audio:** For episode and ad playback.
- **Provider:** Simplifies state management and dependency injection.
- **audio_session:** Manages audio focus and interruptions.
- **youtube_player_flutter:** Embeds YouTube video players in the app.
- **fluttertoast:** Custom toast messages.
- **flutter_rating_bar:** Star-based rating bars.
- **fl_chart:** Charting and data visualization.
- **intl:** Internationalization and date/number formatting.
- **share_plus:** Cross-platform content sharing.
- **image_picker:** Image selection from the gallery.
- **csv:** CSV generation and parsing.
- **file_saver:** Saves files to device storage.
- **http:** Making HTTP requests to connect to the back-end.

3.2.2 Back-end and Tools

- **Node.js + Express:** Main back-end framework for handling RESTful APIs, user authentication, and routing.
- **PostgreSQL:** Relational database used to store user and app data.
- **Cloudinary:** Cloud-based storage service for audio/image storage and streaming.
- **Multer, Streamifier, fluent-ffmpeg:** Used to handle file uploads and extracting audio duration.
- **@ffmpeg-installer/ffmpeg, @ffprobe-installer/ffprobe:** Required for audio metadata processing.
- **OpenAI Whisper:** Used to transcribe podcast audio into text and extract transcript metadata.
- **yt-dlp:** Extracts and downloads audio from YouTube links for external episodes.
- **Dejavu:** Audio fingerprinting and matching of short clips to full episodes (Shazam-like functionality).
- **OpenAI GPT (ChatPot):** Chatbot interface that allows users to summarize, translate, or ask questions about episode scripts using natural language.
- **OpenAI Moderation API:** Automatically filters toxic or harmful user reviews based on language analysis.
- **FastAPI (Python):** Used to build lightweight microservices for recommendation and audio matching.
- **TF-IDF + scikit-learn:** Core of the content-based recommender system using similarity matching.
- **pandas, numpy:** Support data processing and feature extraction in the recommendation microservice.
- **node-cron:** Handles scheduled tasks such as weekly cleanup of old notifications.
- **Postman:** Used for testing backend endpoints during development.
- **Uvicorn:** ASGI server to run FastAPI microservices efficiently.

- **dotenv, jsonwebtoken, bcrypt, cors:** Environment configuration, JWT authentication, password hashing, and CORS handling.

3.3 Database

Athena uses PostgreSQL with clearly separated tables for users, episodes, podcasts, ads, notifications, and interactions (likes, saves, reviews). Relationships are defined using foreign keys. Audio URLs, images, and scripts are stored via Cloudinary and linked to entries.

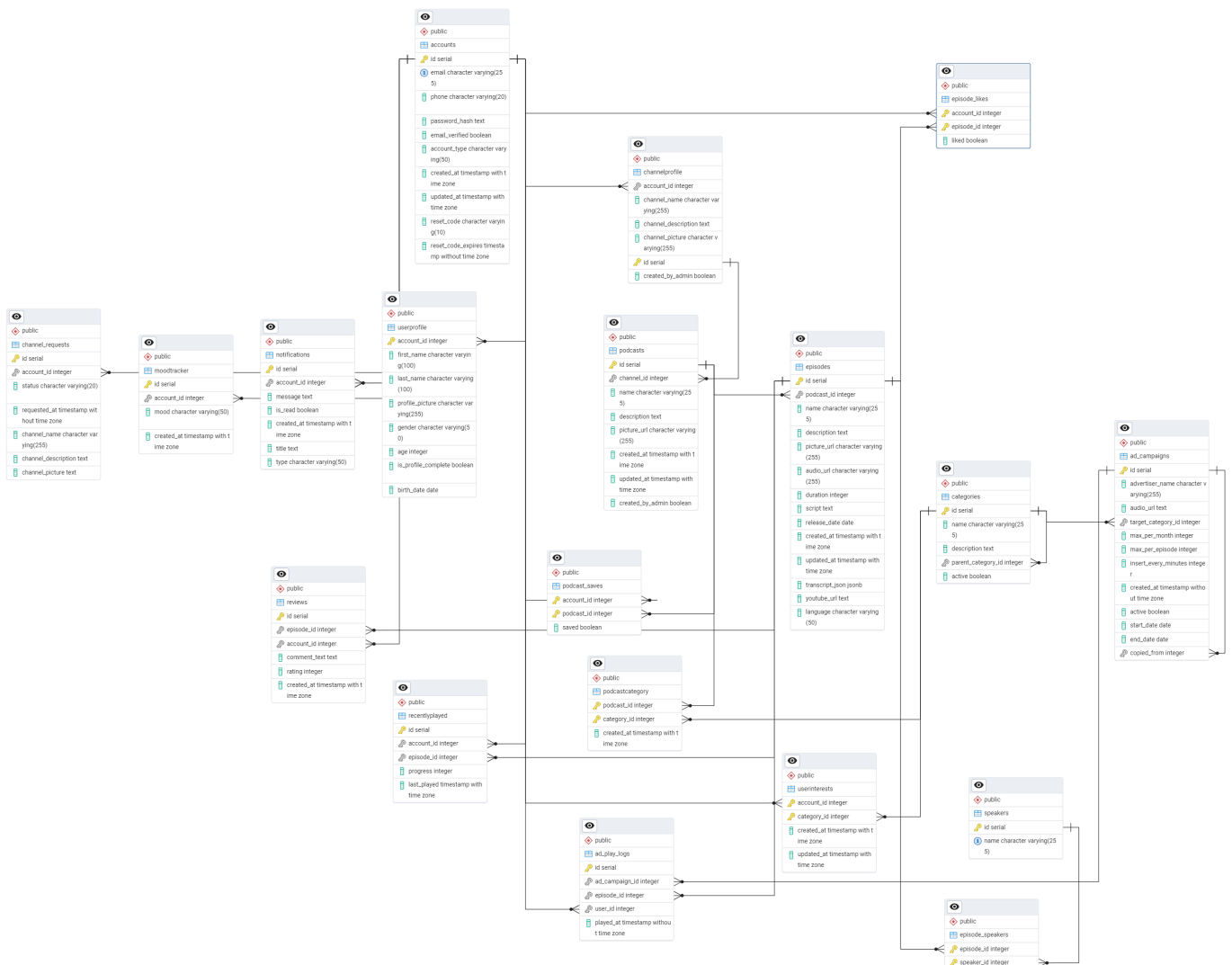


Figure 3.2: ERD of Athena DB

In addition to the main PostgreSQL database, the **Dejavu** audio fingerprinting system uses a separate PostgreSQL database to store fingerprint hashes. This database is queried when matching recorded clips with full episodes.

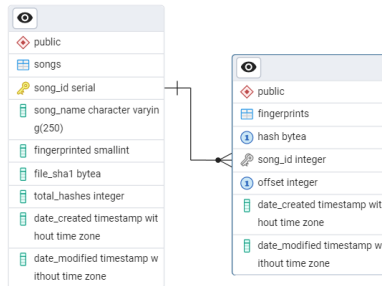


Figure 3.3: ERD of dejavu DB

3.4 Features Implementation

3.4.1 Mobile Application

User Side

- **Registration/Login:** Users can sign up, log in, and reset their passwords. All credentials are securely hashed and verified via email.



Figure 3.4: Splash Screen

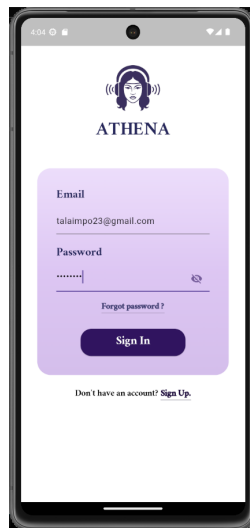


Figure 3.5: Sign In Page

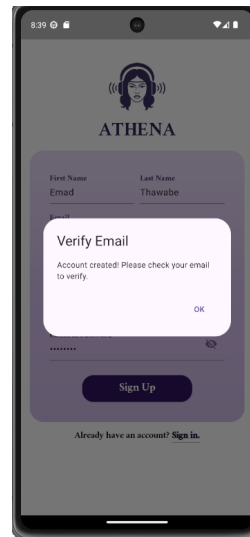


Figure 3.6: Sign Up Page

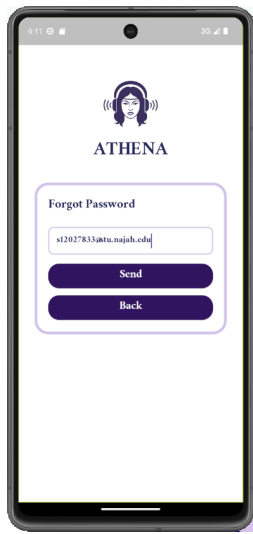


Figure 3.10: Forgot Password

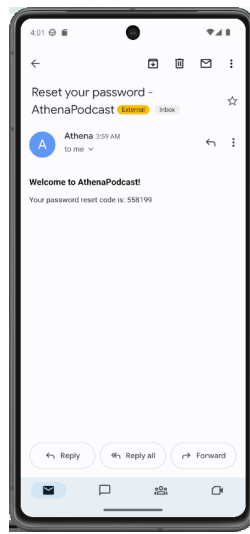


Figure 3.11: Reset code



Figure 3.12: Reset Password

When the user successfully logs in, they will proceed to set up their profile by completing the required information.

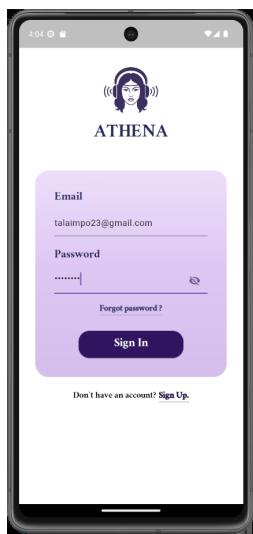


Figure 3.13: User Log-in app successfully



Figure 3.14: Complete profile: Birth date & Gender

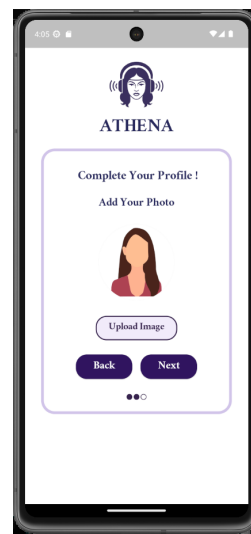


Figure 3.15: Complete profile: Profile Picture

To complete the profile and enter the app, the user must provide their birth date, upload a profile picture, and select the interest categories they like.



Figure 3.16: Complete profile: User interests 1

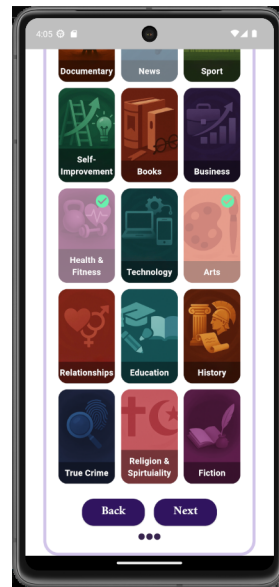


Figure 3.17: Complete profile: User interests 2

Once the user logs in, whether using a new or existing account, they will be directed to the home page, which displays the latest episodes, personalized recommendations, popular channels, recently played episodes, and top categories in the app.



Figure 3.18: Home Page 1

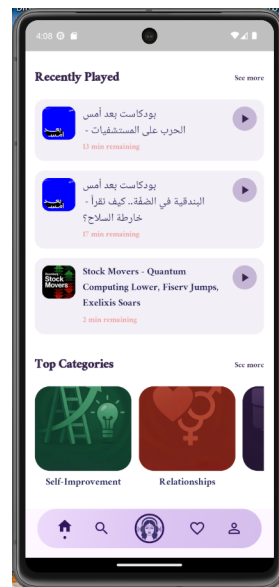


Figure 3.19: Home Page 2

The user can view all items in each section by clicking on "See More". Below is an example of these sections.

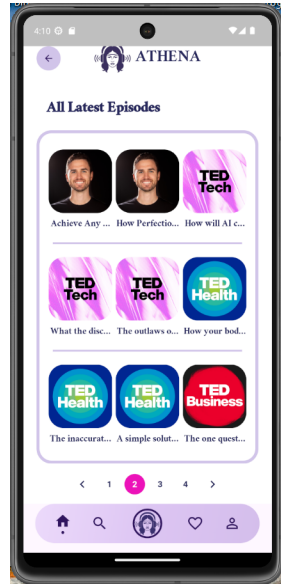


Figure 3.20: All Latest Episodes

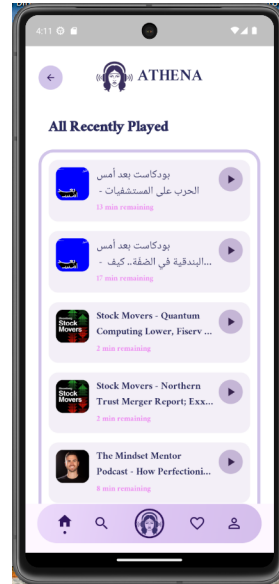


Figure 3.21: All Recently Played

- **Account type conversion:** Users can send a request to convert their account to a channel account, allowing them to share podcasts and episodes. The user profile page allows users to manage their info, view stats, check notifications, and request account conversion.

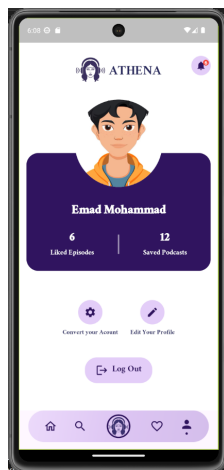


Figure 3.22: User Profile page

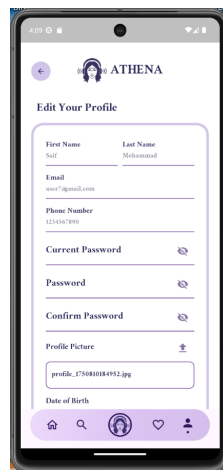


Figure 3.23: User edits profile

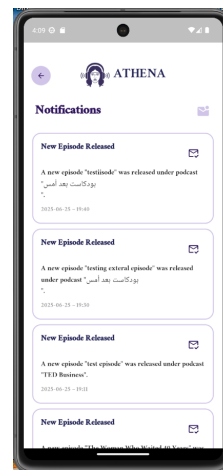


Figure 3.24: User's Notifications

When a user requests to convert their account to a channel, the request remains pending until the admin approves or rejects it. If approved, the account is converted; if rejected, it stays the same. Users cannot submit another request while one is pending and will be notified of the decision.

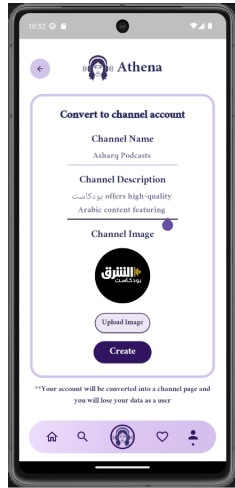


Figure 3.25: The form to convert to a channel account

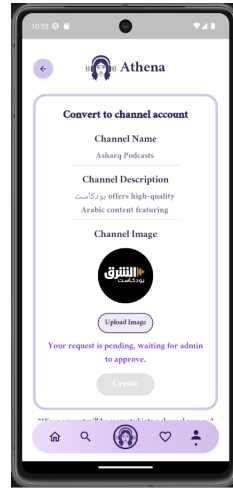


Figure 3.26: Pending Request to Convert to Channel

Every user has a library where they can view their saved podcasts and liked episodes, and they can see all items in each section by clicking on "See All."

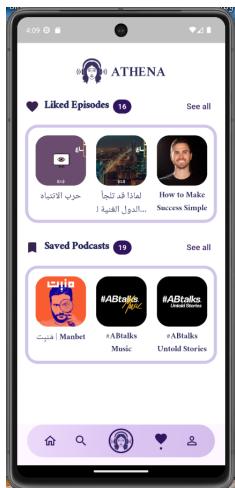


Figure 3.27: Library page

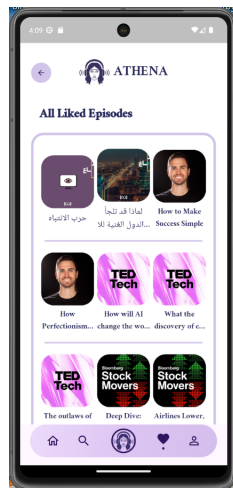


Figure 3.28: All liked episodes



Figure 3.29: All saved podcasts

- **Recommendations:** Individualized recommendations are generated through a dedicated microservice combining TF-IDF vectorization, user interest, and app activity. Users also have **Mood Tracker** within the app, through which users can select their mood from the list below: Happy, sad, loving, depressed, grateful, stressed and angry

When a listener selects a mood, the home page recommendations are automatically updated to be consistent with the current emotional state of the listener. For example, when a listener selects "Stressed," the system looks through their listening history, favorites episodes, and saved podcasts and recommends calming content—such as episodes and podcasts on Health and Fitness or Religion and Spirituality that may help the listener relax.

All recommendations are accessible by a tap on "See More" on the home screen, where episode and podcast recommendations both in their mood and areas of interest are displayed. The recommendations change every time user mood is updated.



Figure 3.30: Recommendations (Before Select Mood)



Figure 3.31: Mood Tracker – User Select the Current Mood

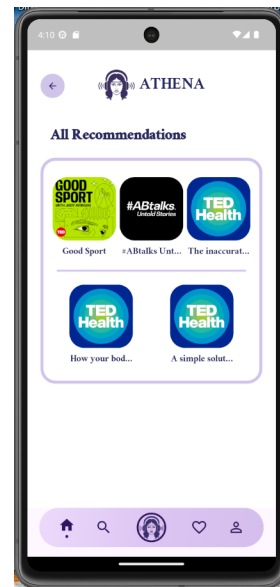


Figure 3.32: Updated Recommendations Based on Mood

- **Channels, Podcasts, and Episodes:** Users can browse all channels, explore their podcasts, and view detailed episode lists with images, descriptions, and playback options.

As indicated on the homepage, there is a list of the most popular channels. Clicking on "See More" leads to the **All Channels** page that displays two groups of channels:

- **Regular Channels:** Created by registered users who develop their own podcasts and episodes.
- **External Channels:** Contain imported podcasts from YouTube.

Users can search for channels using a text search box, as well as by category and language filters.

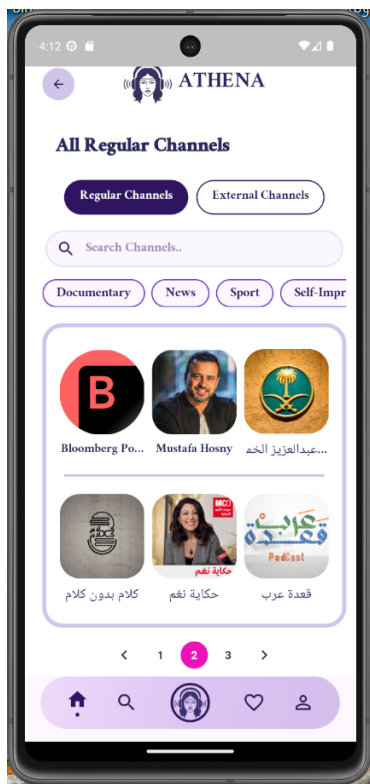


Figure 3.33: All Regular Channels

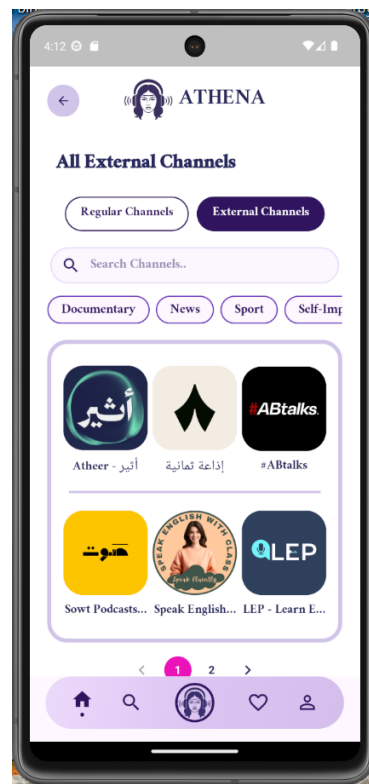


Figure 3.34: All External Channels

When a channel is selected, either from the Home Page or the All Channels page, they are taken to the Channel Details page. The page shows the name of the channel, whether it is a regular or external channel, a description, and a list of all podcasts in the channel. Users can also search for individual podcasts on the selected channel.



Figure 3.35: Channel Page With Details

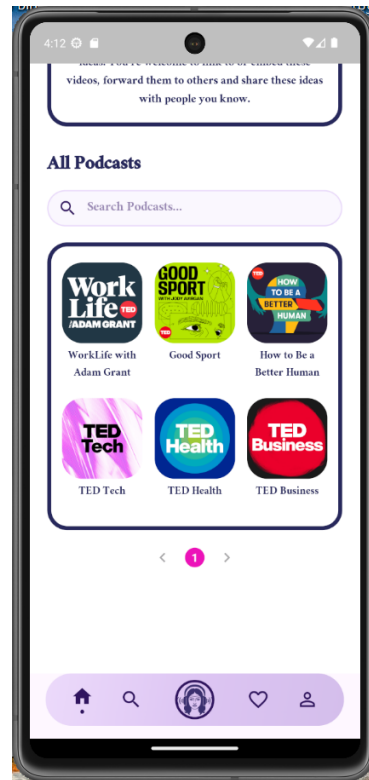


Figure 3.36: All Podcasts in Channel

Upon a user's selection of a podcast, either from anywhere within the app or from a channel page, the Podcast Page is displayed. Information on the podcast, including its title, description, languages, and podcasts' category, whether regular or external podcast. It also shows the number of users who have bookmarked the podcast and the number of episodes under it.

All the episodes that are included in the podcast are available here and one can search for specific episodes in the list that they would like to view. One can save the podcast as well in order to watch it in their library, or remove the podcast save.



Figure 3.37: Recommendations (Before Select Mood)



Figure 3.38: Mood Tracker – User Select the Current Mood

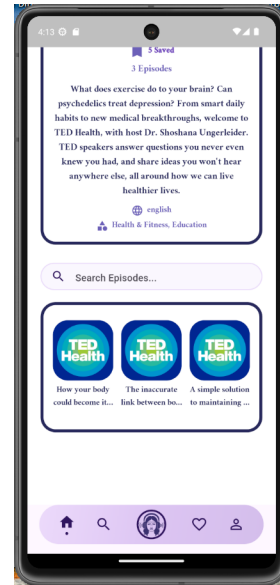


Figure 3.39: Updated Recommendations Based on Mood

Upon a user’s selection of an episode, either from anywhere within the app or from a podcast page, the Episode Page is displayed. Information about the episode, including its title, description, language, speakers, and duration of the episode. The page also shows the number of users who have liked the episode, the reviews made by the users, and the average rating of the episode. Users can like the episode and submit a review and the episode will be added to their library.

As mentioned above, there are two types of channels: **Regular** and **External**. Regular channels are created by individuals who upload their own content, while external channels consist of podcast episodes imported from YouTube.

Episodes under **External Channels** display an embedded YouTube video player, allowing users to view and listen to the episode directly from YouTube. Alternatively, users can choose to listen to the audio only.

In return, episodes under **Regular Channels** provide audio playback only.

Both types of episodes include a real-time transcript. The transcript behavior and layout are specified separately for each episode type in the following sections.

External Episode Page

On the External Episode Page, Users can watch or listen to the episode and view a real-time transcript. They can also like and review it.



Figure 3.40: External Episode 1

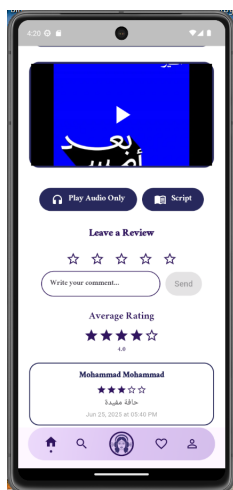


Figure 3.41: External Episode 2

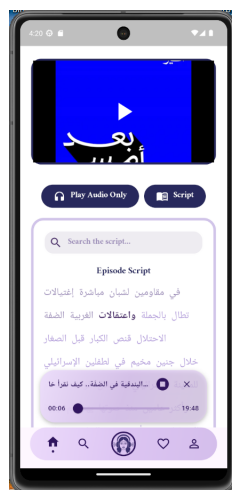


Figure 3.42: Listen External Episode

Regular Episode Page

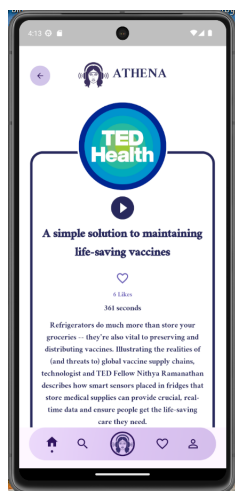


Figure 3.43: Regular Episode 1

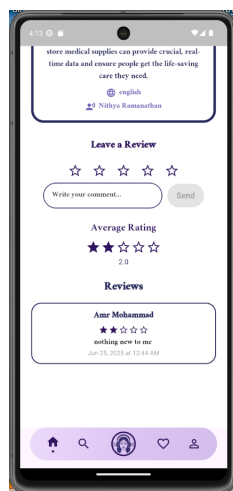


Figure 3.44: Regular Episode 2

- **Review Moderation:** When a user submits a review, Athena automatically filters out harmful or inappropriate content using OpenAI’s moderation tools and a custom blacklist. Users can still post reviews and rate episodes as long as the content is appropriate, and they also have the option to delete their own reviews at any time.

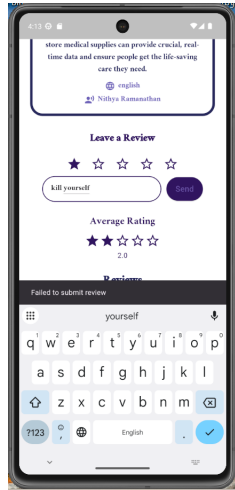


Figure 3.45: Bad Review Rejected

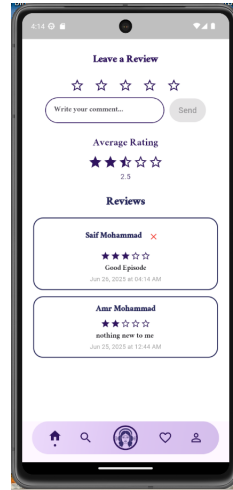


Figure 3.46: Good Review Submitted

Listen Page (Regular Episodes)

If a listener selects to play a regular episode, the listener is directed to the **Listen Page**. Listeners on the page can change playback speed, enable a sleep timer, and share the episode with their friends.

Upon entering the listen page, the user is automatically matched with the most suitable audio ad from among the active ad campaigns. The selected ad plays during the episode according to its predetermined rules.

There is also a real-time transcript being displayed while the episode plays, and users can interact with a built-in chatbot to discuss the content.



Figure 3.47: Listen Regular Episode

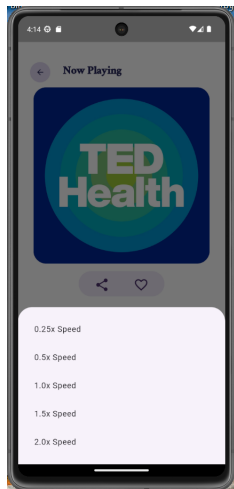


Figure 3.48: Audio Playback Speed

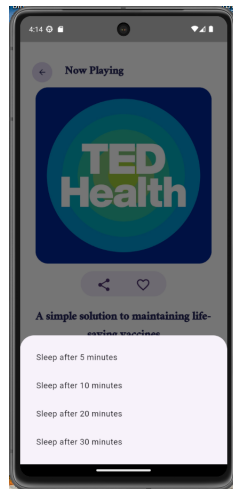


Figure 3.49: Set Sleep Timer

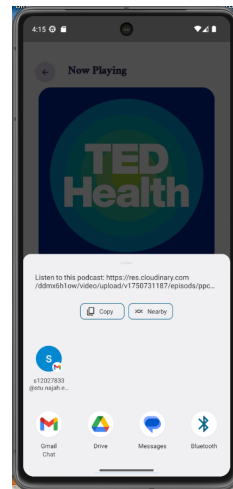


Figure 3.50: Share Episode

- **Transcription:** Each episode is transcribed using OpenAI Whisper. Transcripts are displayed word-by-word with real-time highlighting and timestamps.



Figure 3.51: Episode Transcript

- **Chatbot:** A GPT-based assistant allows users to ask questions, request summaries, get script translations, identify speakers, define terms, compare episodes, understand mood, receive listening recommendations, and more.



Figure 3.52: Chatbot 1



Figure 3.53: Chatbot 2



Figure 3.54: Chatbot 3

Search Page

The Search Page may also be accessed by tapping the search icon from the bottom navigation bar or by clicking "See More" in the Top Categories section on the Home Page. On this page, users can browse all the categories that are available. If a category is clicked, the user is taken to the Category Page, where all podcasts in the chosen category are listed along with a featured episode.

Users can also do text-based searches across the whole app for channels, podcasts, and episodes. The search supports filters by category and language and includes smart suggestions and autofill for easier use.

The app also includes a Shazam-like audio search feature where users are able to search up and open an episode by recording a short audio clip.

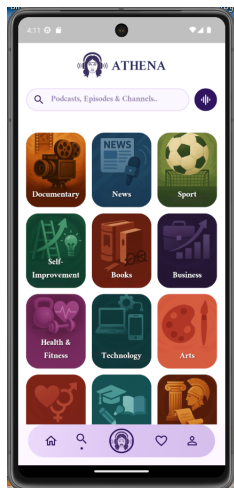


Figure 3.55: Search & All Categories

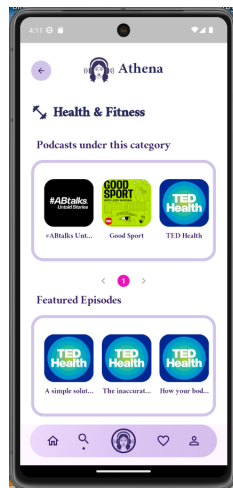


Figure 3.56: Category Page

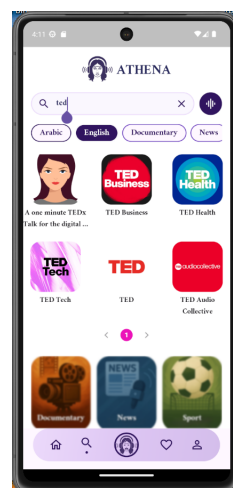


Figure 3.57: Searching

- **Audio Matching:** Users can record short audio clips, which are matched to episodes using the Dejavu fingerprinting microservice.



Figure 3.58: Record Short Audio Clip

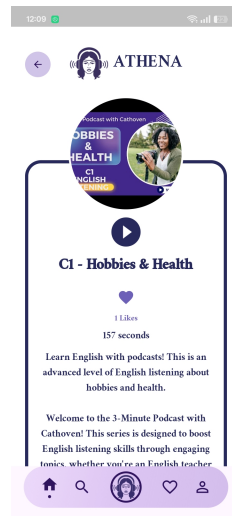


Figure 3.59: The Result Of Matching

Channel Side

- **Channel Profile:** When a user's request to convert their account to a channel is approved, they are redirected to the channel version of the app upon login.

In the Channel Profile page, users can view both their channel information and personal details, such as channel name, description, and analytics about their content, including the number of episodes and podcasts they own, as well as total likes and saves received from other users. They can also manage and edit this information. Additionally, the channel user can view their notifications and mark them as read individually or all at once.



Figure 3.60: Channel Profile 1

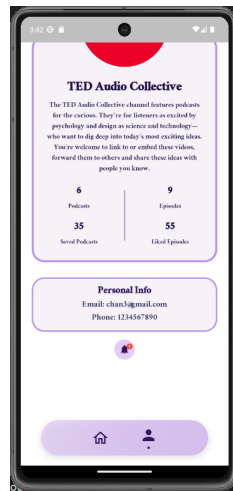


Figure 3.61: Channel Profile 2

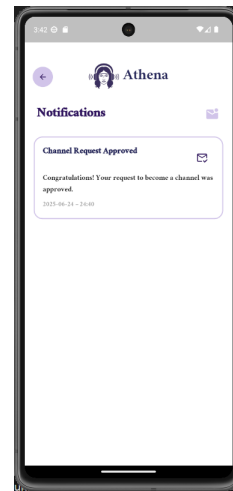


Figure 3.62: Channel Notifications

- Podcast Creation:** Podcasts may be created and organized by channels. When the user logs into the channel, he is brought to the Home Page, where he can view all his podcasts, search for them, add a new one, or delete an existing one.

To add a new one, the user must enter the title, description of the podcast, upload the image, and select one or several categories.

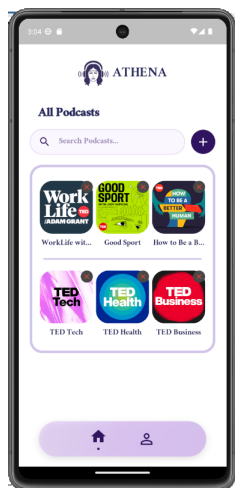


Figure 3.63: All Podcasts

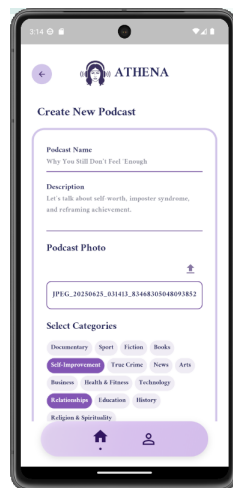


Figure 3.64: Create New Podcast

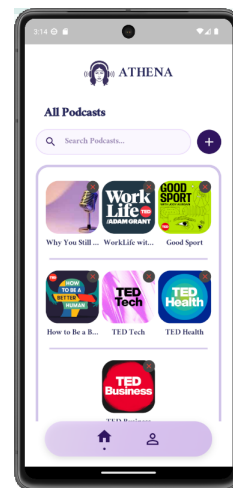


Figure 3.65: Podcast Created

When a channel selects a specific podcast, all related infor-

mation is displayed, including the podcast's name, number of saves, episode count, description, languages, and categories. The channel can also view all episodes under the podcast, search within them, delete existing episodes, or add new ones.

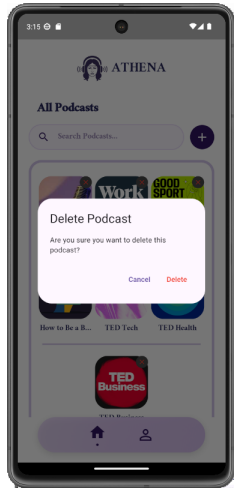


Figure 3.66: Delete Podcast



Figure 3.67: Podcast Page

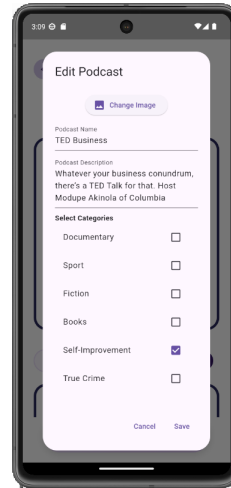


Figure 3.68: Edit Podcast

- **Episode Creation:** Channels can publish new episodes by providing audio, an image, episode title, description, language, release date, and speaker names. The release date and speaker names are optional.

Once a post of an episode is clicked, a wait message is shown while background processes are started. These include auto-transcription and OpenAI Whisper-based JSON transcription generation as well as audio duration extraction. If description or language fields are left blank, Whisper will attempt to automatically fill them out.

If the audio file size is greater than the maximum (25MB), it's compressed or split into 5-minute parts to prevent slowdown on Whisper processing. The image and sound are uploaded and stored in Cloundinary.

Moreover, a one-time audio fingerprint and hash are generated for the episode by Dejavu so that it can be recognized in the future.



Figure 3.69: Create New Episode 1

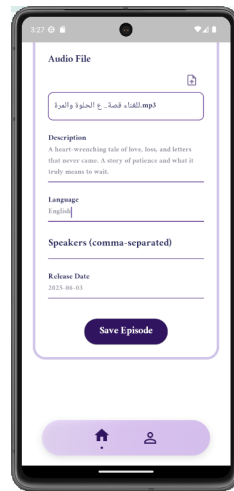


Figure 3.70: Create New Episode 2



Figure 3.71: Wait Message

If a channel selects an episode, they will be brought to the Episode Page, where they can view all the episode details, listen to the audio, and read all the reviews written by users.

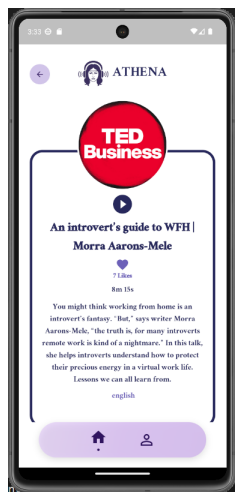


Figure 3.72: Episode Page.1

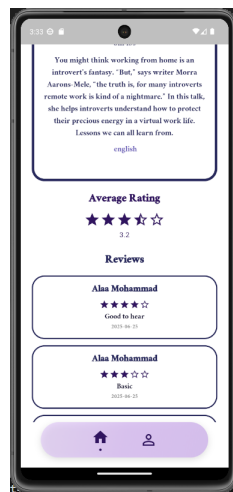


Figure 3.73: Episode Page.2

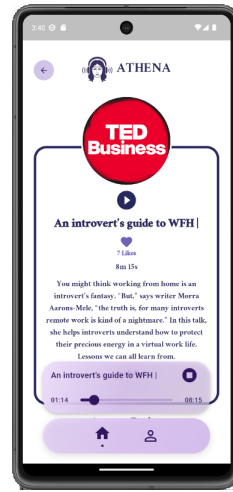


Figure 3.74: Listen Episode

Channel can also edit the episode's information or delete the selected episode.

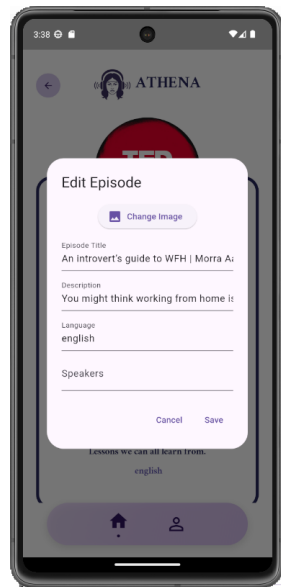


Figure 3.75: Edit Episode

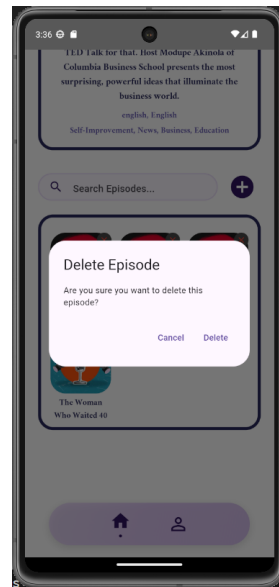


Figure 3.76: Delete Episode

Admin Side

- **External Channel Creation:** After the admin logs into the app, the admin is directed to the Home Page, which contains a list of all external channels. The admin can search the channels textually or using filters, add a new external channel, or delete an existing one.

In the case of adding a new channel, the admin has to provide the channel's name, description, and image. In the case of selecting a specific channel, the admin is able to view all its details as well as the podcasts within it. The admin is able to search for podcasts, add new podcasts, or delete any podcast from the channel.

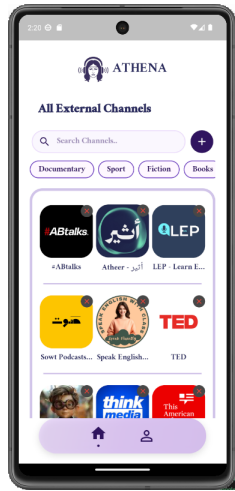


Figure 3.77: All External Channels

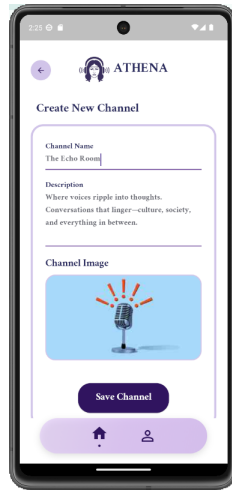


Figure 3.78: Create External Channel



Figure 3.79: External Channel Page

- External Podcast Creation:** To add a new external podcast, the admin must enter the title, description of the podcast, upload the image, and select one or several categories. When the admin selects a specific podcast, all related information is displayed, including the podcast name, episode count, and description. Also the admin can view all episodes under the podcast, search within them, delete existing episodes, or add new ones.

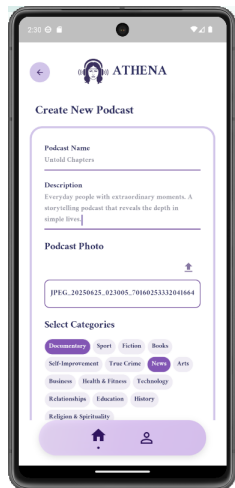


Figure 3.80: Create External Podcast 1

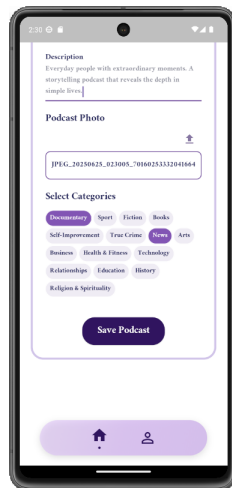


Figure 3.81: Create External Podcast 2

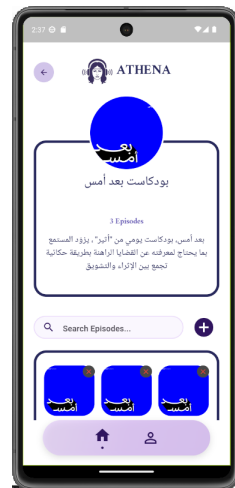


Figure 3.82: External Podcast Page

- **External Episode Creation:** When the admin chooses to create a new external episode, he has to provide details such as the episode name, image, description, YouTube link, language, speaker names, and release date.

This process works similarly to how channels create regular episodes, with the main difference being that instead of uploading an audio file, the admin provides a YouTube link. The system uses `yt-dlp` to extract the audio from the YouTube video, uploads it along with the image to Cloudinary, and generates a transcript using OpenAI Whisper. An audio fingerprint is also created using Dejavu.

If the name or description fields are left empty, they are automatically extracted from the YouTube video. If the language is not provided, it is detected by Whisper. Speaker names and release date are optional and can be left blank.

Once the episode is created, the admin can click on it to view all its details and either watch the video or listen to the extracted audio.

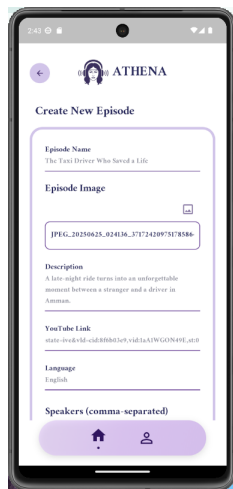


Figure 3.83: Create External Episode 1

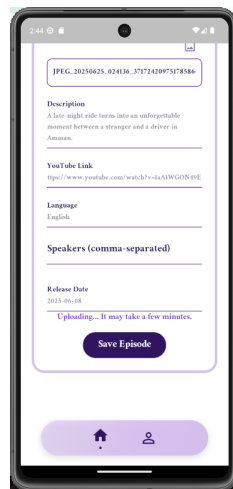


Figure 3.84: Create External Episode 2



Figure 3.85: External Episode Page

- **Admin Profile:** When the admin goes to the Profile Page, the admin can view his account details and edit them as needed.

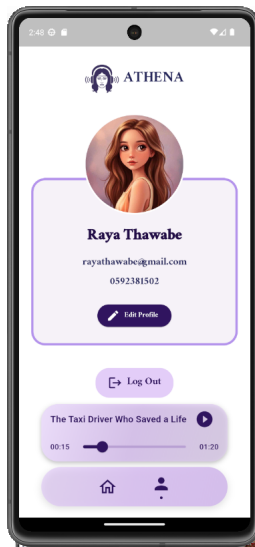


Figure 3.86: Admin Profile

3.4.2 Web Dashboard

The Web dashboard is designed exclusively for the admin to manage the application.

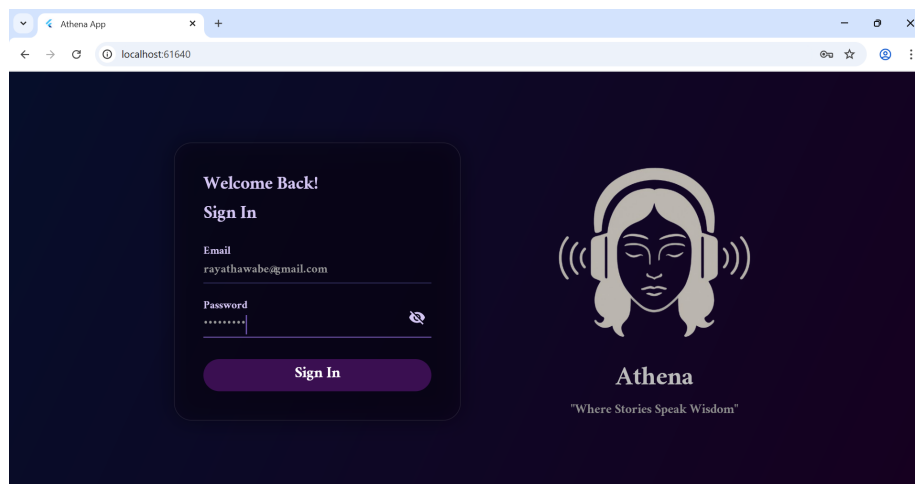


Figure 3.87: Login Page

- **All Users:** The admin can view a table of all users in the application, with the ability to search for users, view their details and delete their accounts.

No.	Full Name	Email	Phone Number	Gender	Age	Actions
1	Emad Thawabe	s12027833@ta.najah.edu	0592381502	Male	23	
2	Tala AbuSoud	talaipo23@gmail.com	0597474949	N/A	N/A	
3	Ola Ahmad	user20@gmail.com	1234567890	Female	25	
4	Tuleen Ahmad	user19@gmail.com	1234567890	Female	25	
5	Deema Ahmad	user18@gmail.com	1234567890	Female	25	
6	Raya Ahmad	user17@gmail.com	1234567890	Female	25	

Figure 3.88: All Users Table

If the admin selects a specific user, they can view the user's account details, including the total number of likes and saved items, user reviews, listening time percentage by category, and the total listening time in the app.

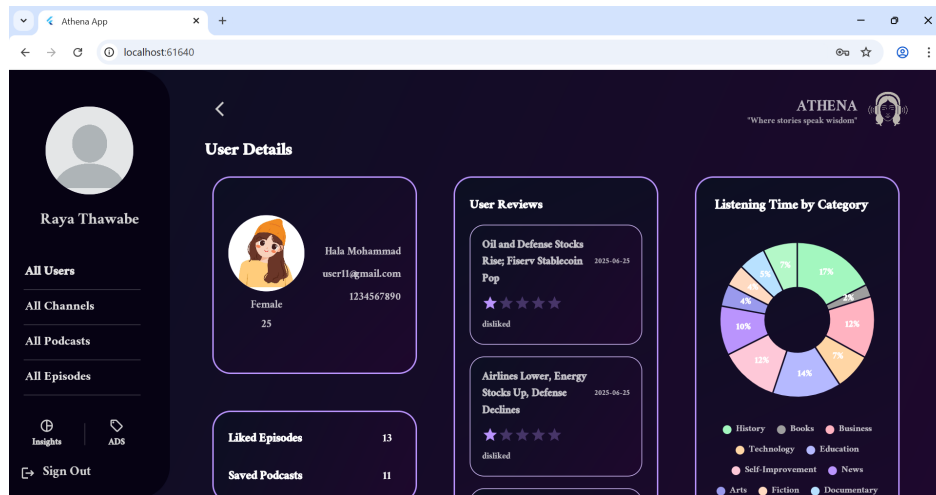


Figure 3.89: User Details Part.1

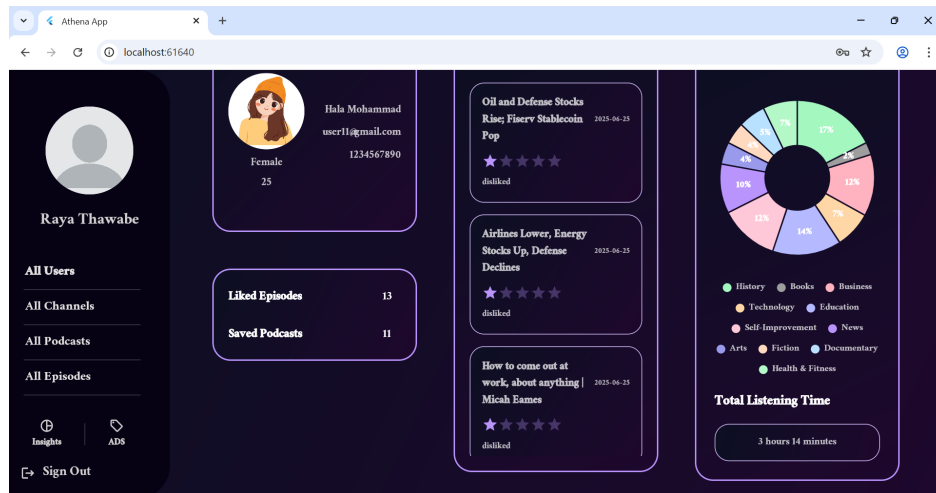


Figure 3.90: User Details Part.2

- **All Channels:** In this tab, the admin gets to view four tables related to channel management. The admin is able to view some channels, search in the tables, or remove a certain channel.
 1. **All Requested Channels:** Displays all user requests to transform their accounts into channels, pending, approved and rejected requests.

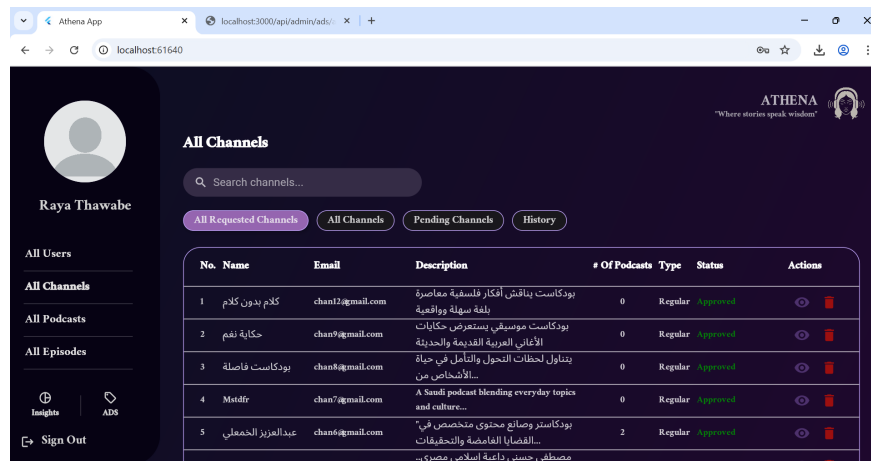


Figure 3.91: All Requested Channels Part.1

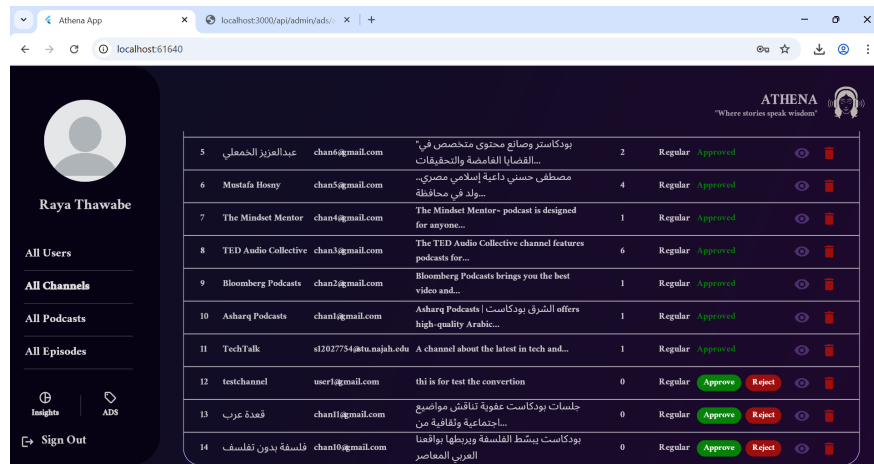


Figure 3.92: All Requested Channels Part.2

2. **All Channels:** Shows all channels in the app with their details, including their the type, regular or external channel.

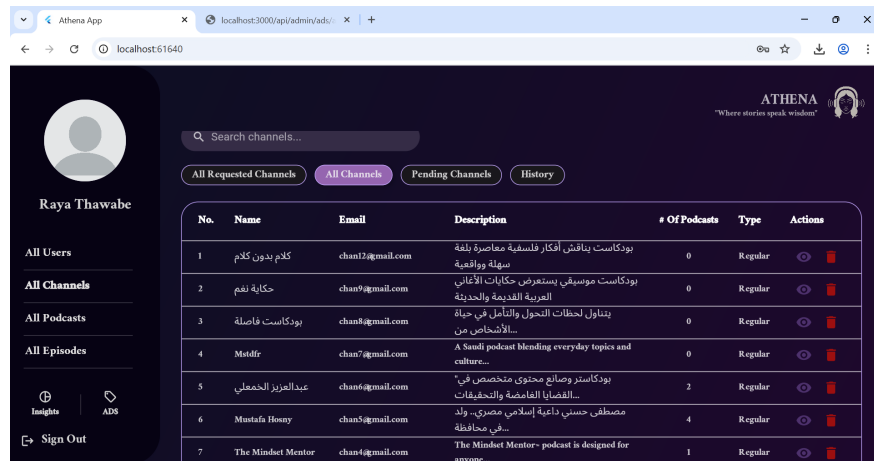


Figure 3.93: All Channels

3. **Pending Channels:** Lists all pending channel requests waiting for admin approval or rejection.

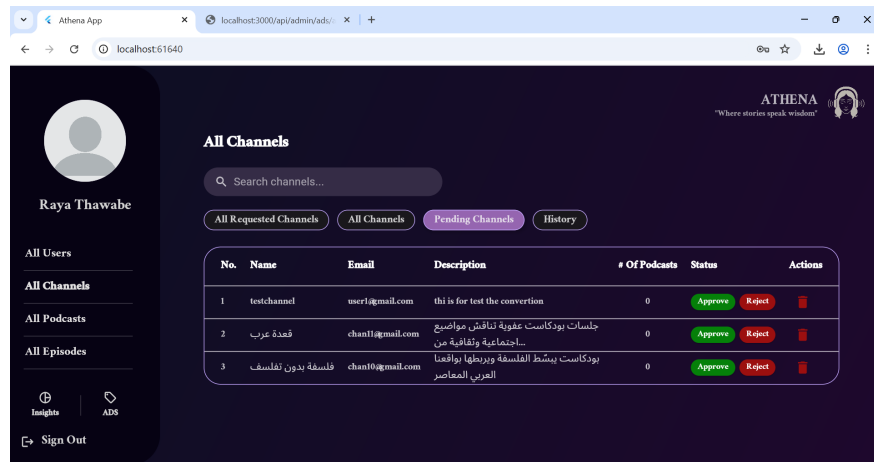


Figure 3.94: All Pending Channels

4. **History:** Contains a record of all channel requests that have already been approved or rejected by the admin.

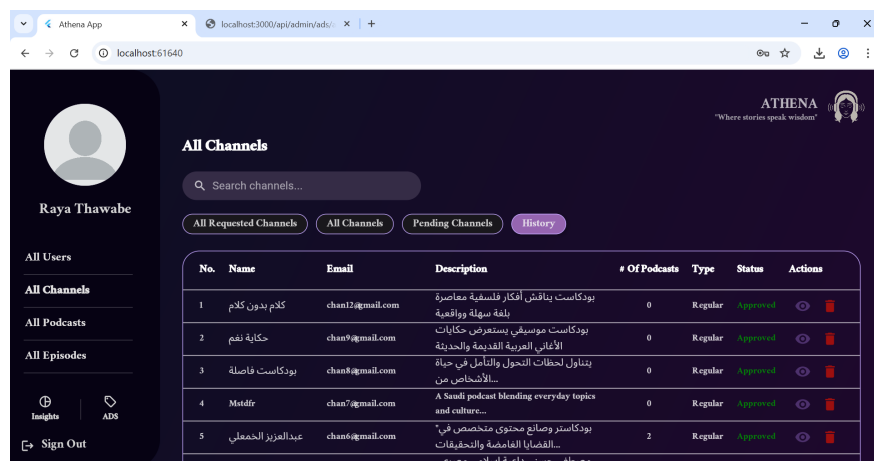


Figure 3.95: History Part.1

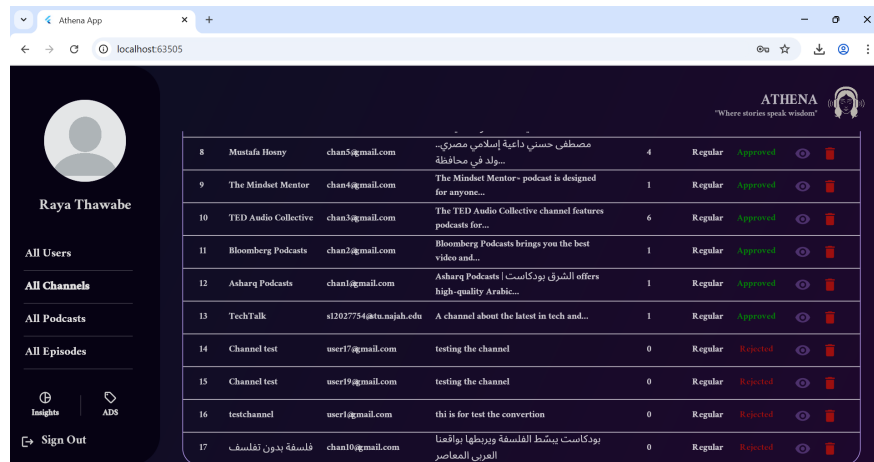


Figure 3.96: History Part.2

If the admin chooses to view a specific channel, he is taken to the Channel Details Page. The page contains the details of the channel, the creation time, and its podcast list. The admin can click on any podcast for more information, and the page shows the total number of podcasts and episodes. There is also a button to view the details of the channel owner.

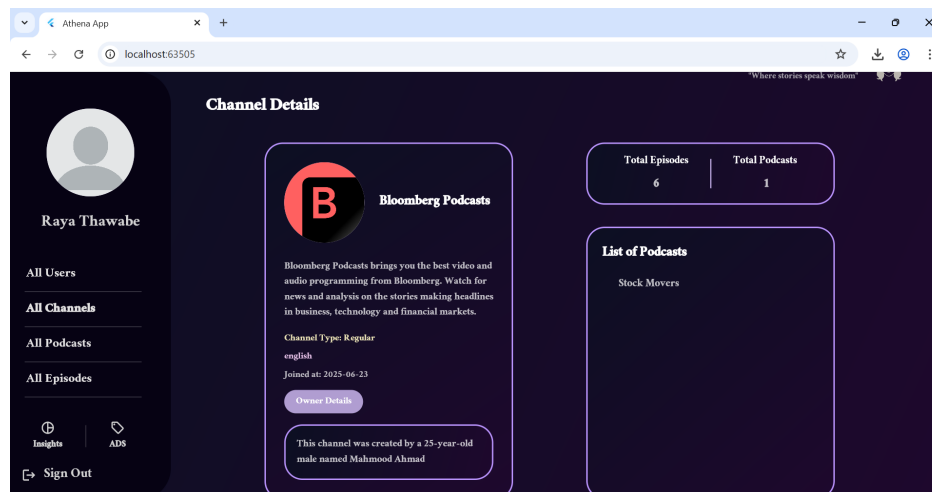


Figure 3.97: Channel Details

- **All Podcasts:** In this tab, the admin is also permitted to view all podcasts in the app along with their details, i.e., their type and the channel they belong to. The admin is also allowed to search for any specific podcast, view their details, or delete them if needed.

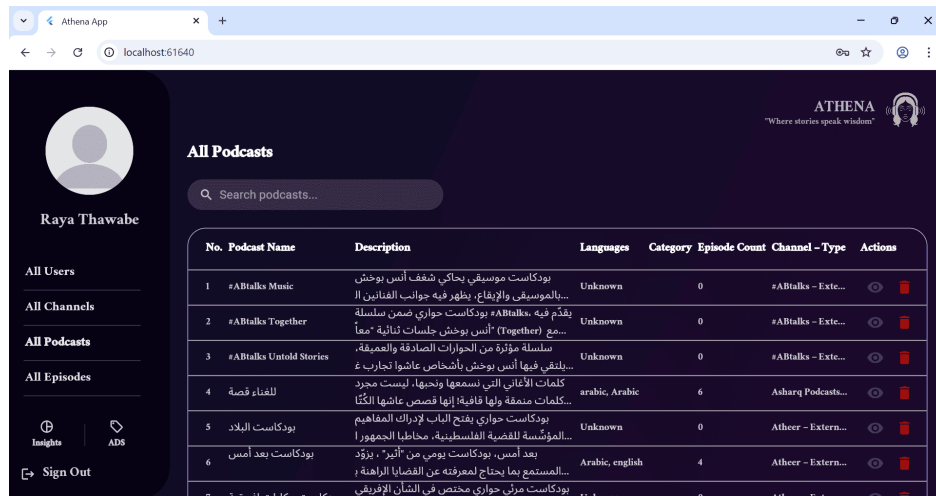


Figure 3.98: All Podcasts

If the admin wants to view a podcast, the **Podcast Details Page** is shown. It contains all data about the podcast, including its categories, episodes, and saves total. A list of episodes that the podcast contains is also presented, and the admin can click on whichever episode to view its full data.

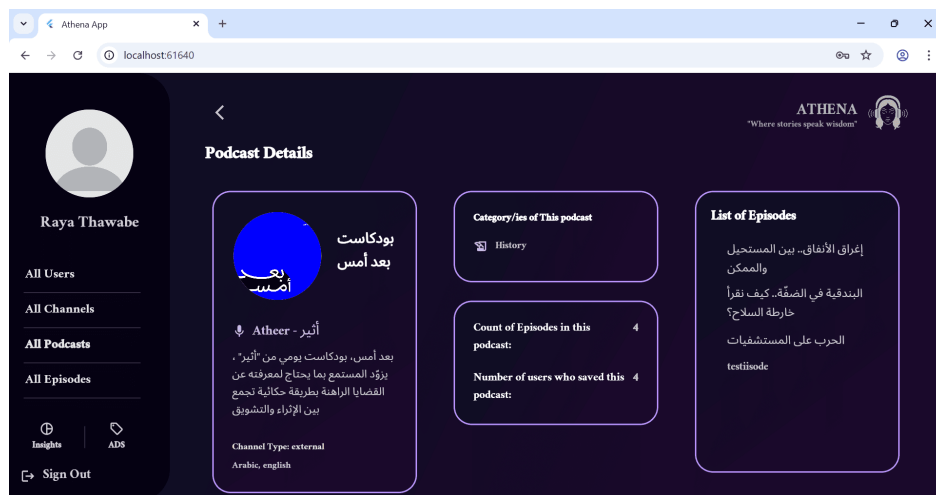


Figure 3.99: Podcast Details

- **All Episodes:** In this tab, the admin is able to view all the episodes in the app along with their details, i.e., the episode type and the channel and the podcast related to it. The admin

can search for a specific episode, view its details, or delete it if necessary.

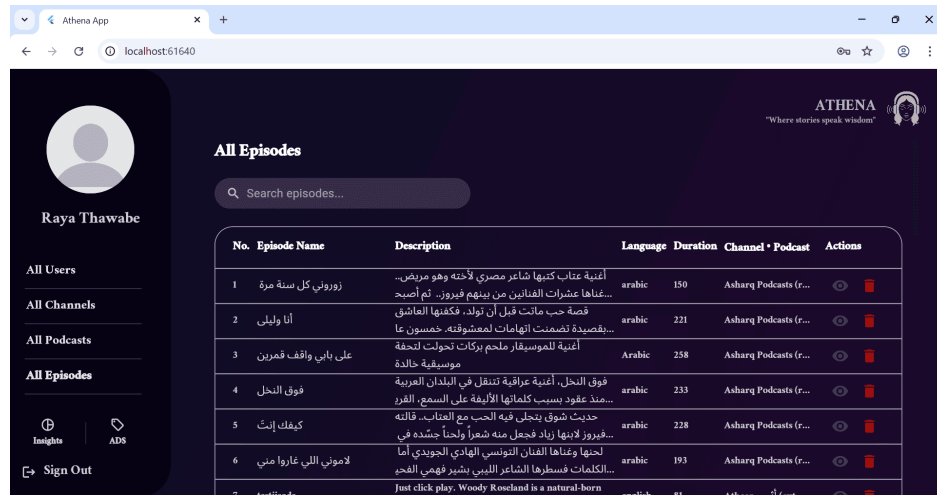


Figure 3.100: All Episodes

When the admin selects an episode to view, they are taken to the **Episode Details Page** where all information related to the episode, including user reviews and the average rating, are displayed. There are two buttons on the page: one that displays the list of speakers and the other that displays the episode script.

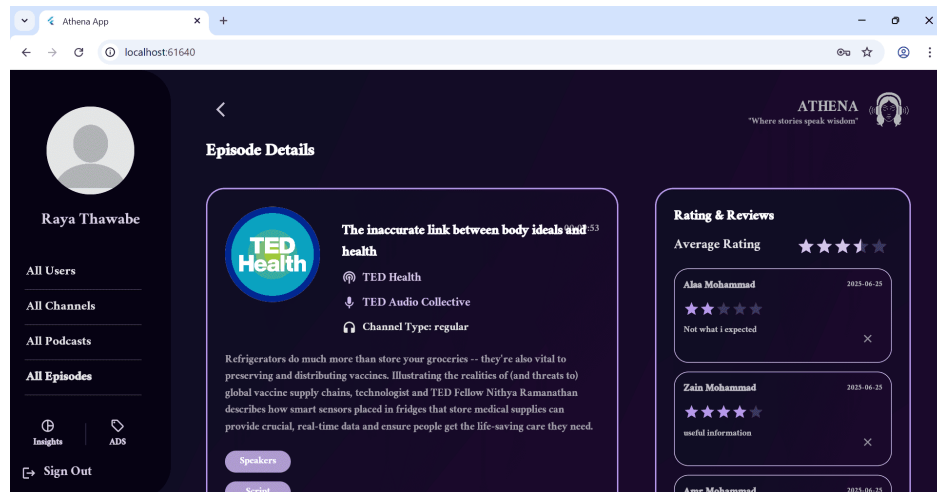


Figure 3.101: Episode Details

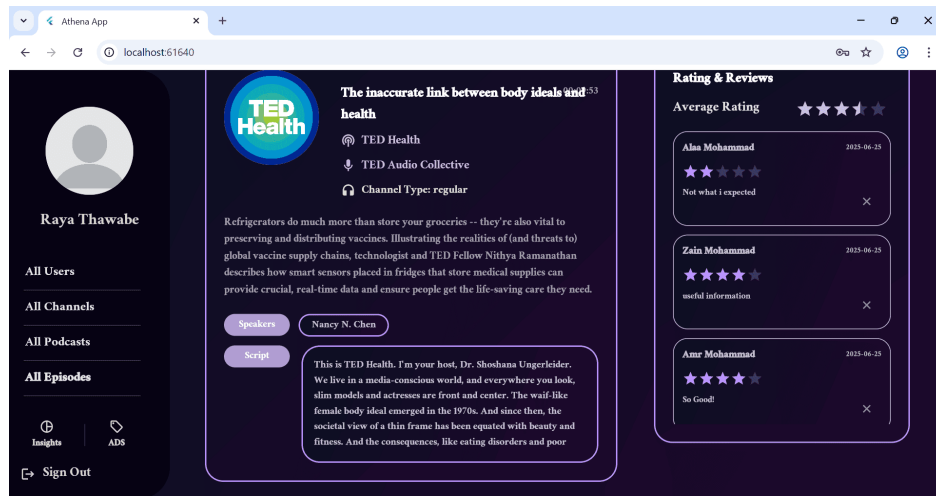


Figure 3.102: Episode Speakers and Script

- **Insights:** This section provides the admin with general insights and analytics based on all data within the app.

First, the admin sees a **Highlights** section that displays:

- Top liked episode
- Top saved podcast
- Top listener with their total listening time
- Most popular category

Next, several charts display the **Top 5 Rankings**, including:

- Top 5 liked episodes
- Top 5 listeners
- Top 5 saved podcasts
- Top 5 categories
- Top 5 rated episodes
- Top 5 most reviewed episodes

Following that is the **Activity Snapshot**, which includes:

- Most inactive users
- New users (last month)
- New episodes (last month)
- Average listening time per user

Finally, the **Total Counts Dashboard** displays overall statistics such as:

- Total number of users, channels, podcasts, and episodes
- Total listening hours
- Average episode duration

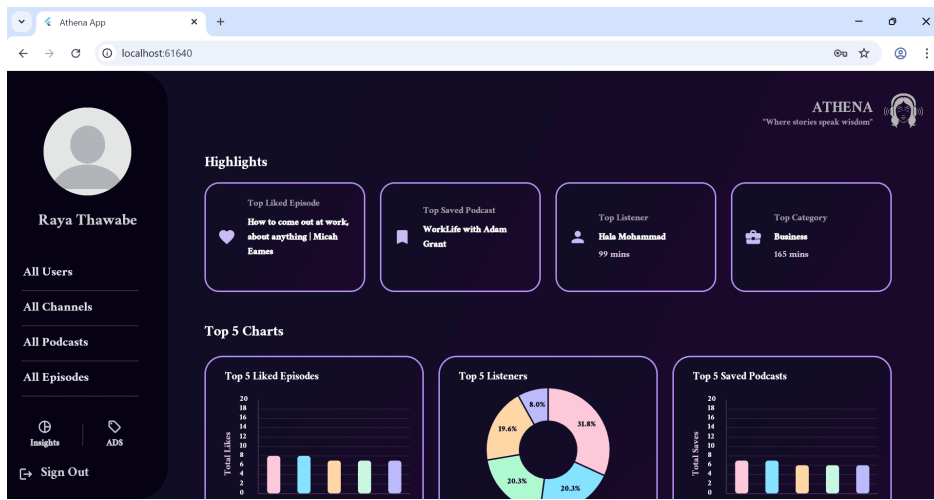


Figure 3.103: Insights Part.1

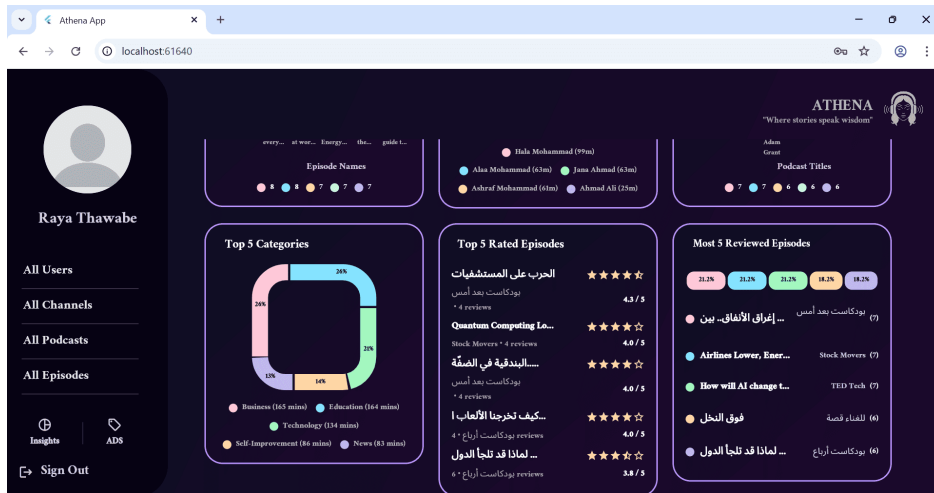


Figure 3.104: Insights Part.2

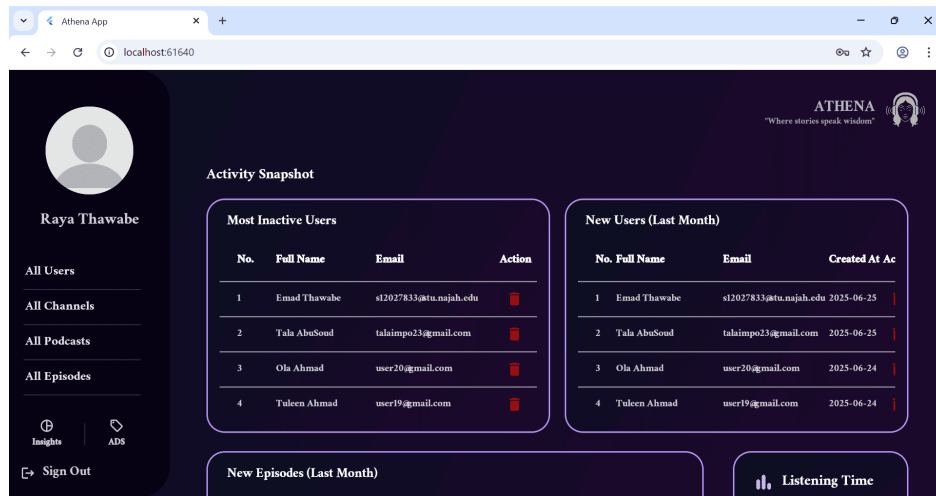


Figure 3.105: Insights Part.3

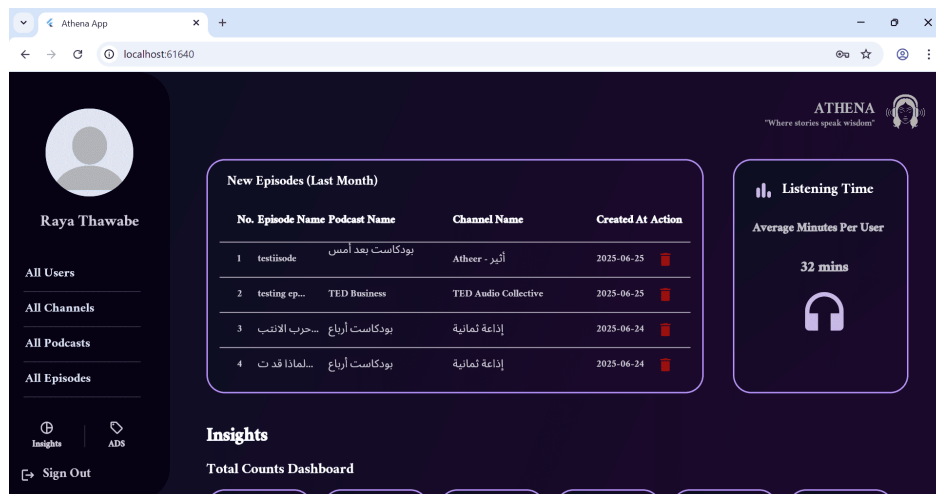


Figure 3.106: Insights Part.4

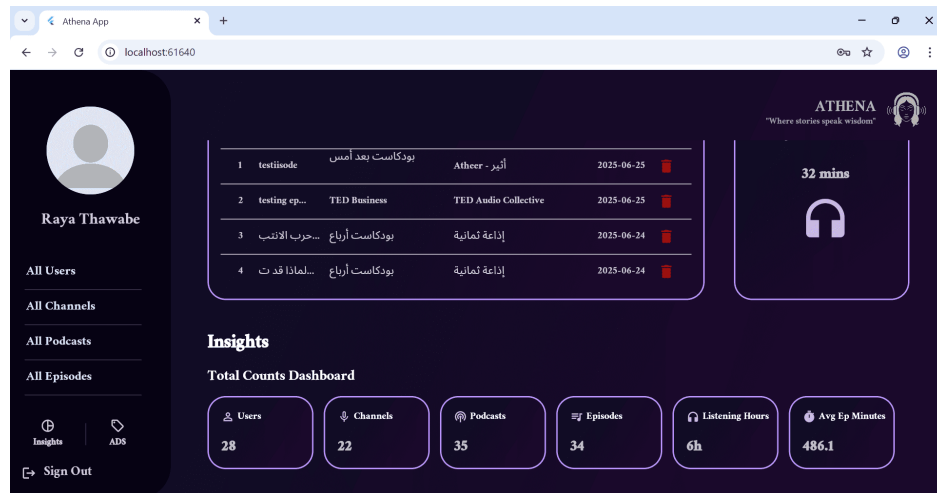


Figure 3.107: Insights Part.5

- **Ads Manager:** The Ads Manager allows the admin to view insights related to all ad campaigns and perform actions such as creating, managing, updating, and deleting campaigns.

The section begins with a **Total Counts** summary, showing:

- Total campaigns
- Active campaigns
- Expired campaigns
- Monthly ad plays

The admin can also view a chart of the top-performing ads based on user reach.

To create a new campaign, the admin must enter the advertiser's name and upload the ad audio. They can then define campaign rules, including:

- Maximum plays per month
- Maximum plays per episode
- Interval (in minutes) between ad insertions
- Optional targeting of a specific category or keeping it general
- Optional start and end dates to schedule or limit the campaign duration

All campaigns are listed in a table showing their details, rules, and active status. The admin can view, edit, or delete any campaign. Expired campaigns can also be renewed if needed.

Additionally, the admin can download all campaign data as a CSV file for deeper analysis and reporting.

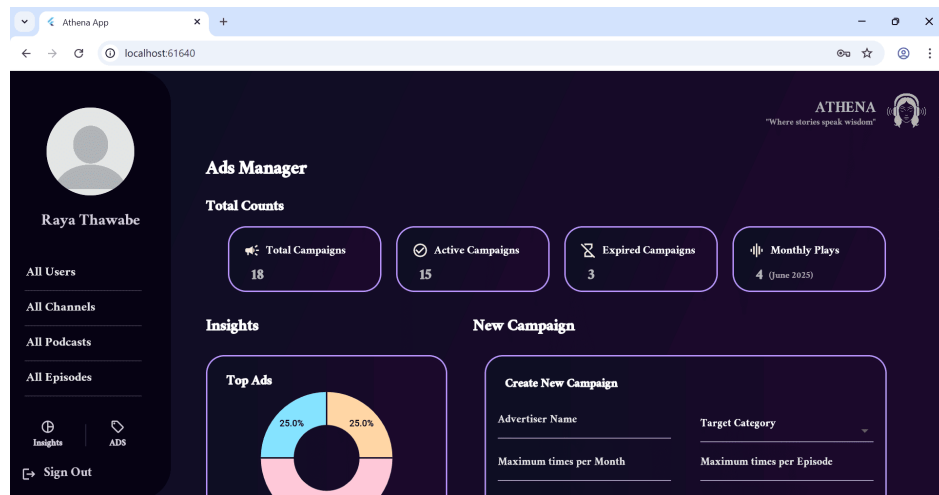


Figure 3.108: Ads Manager Part.1

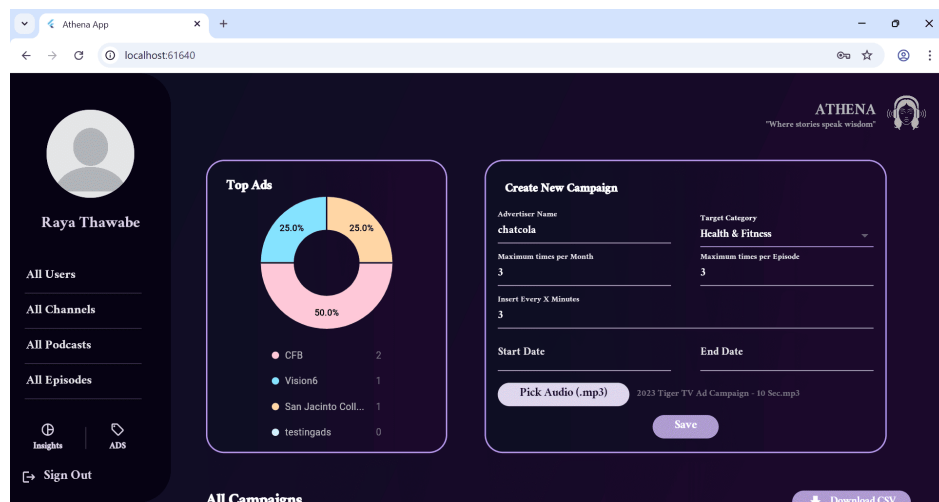


Figure 3.109: Ads Manager Part.2

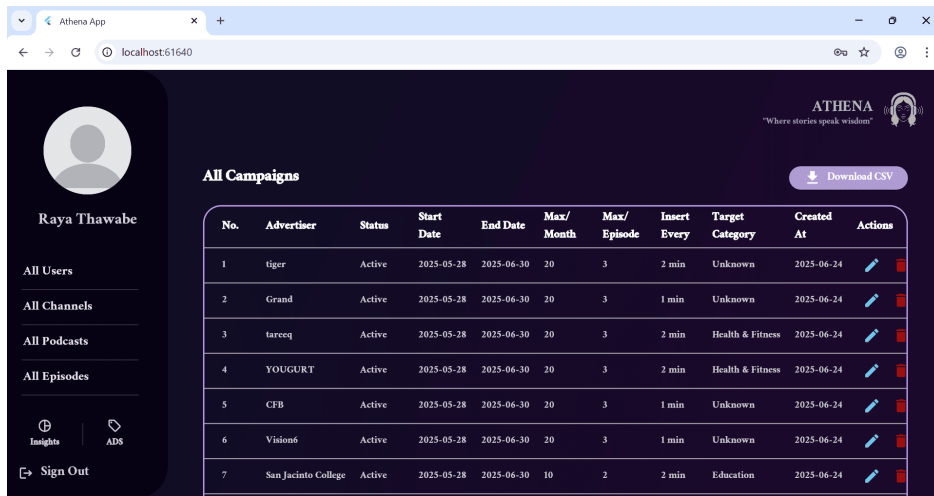


Figure 3.110: All Campaigns - Ads Manager Part.3

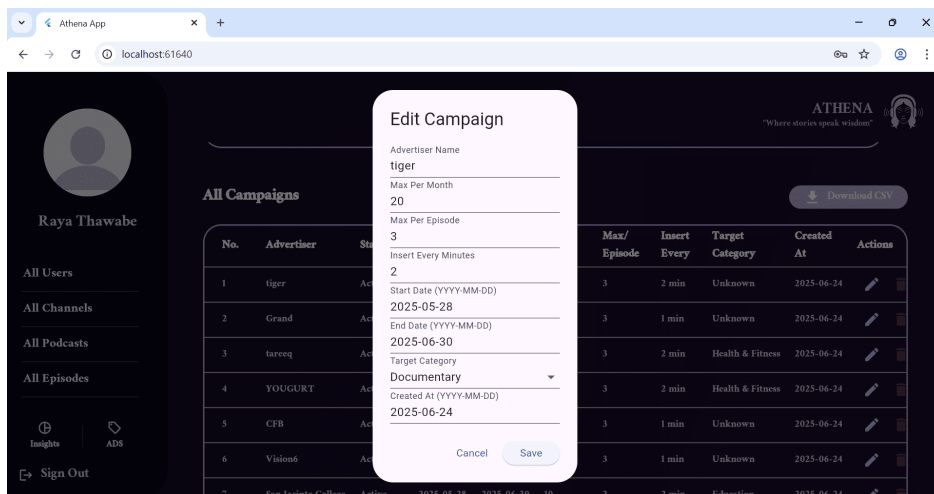


Figure 3.111: Edit Campaign - Ads Manager Part.4

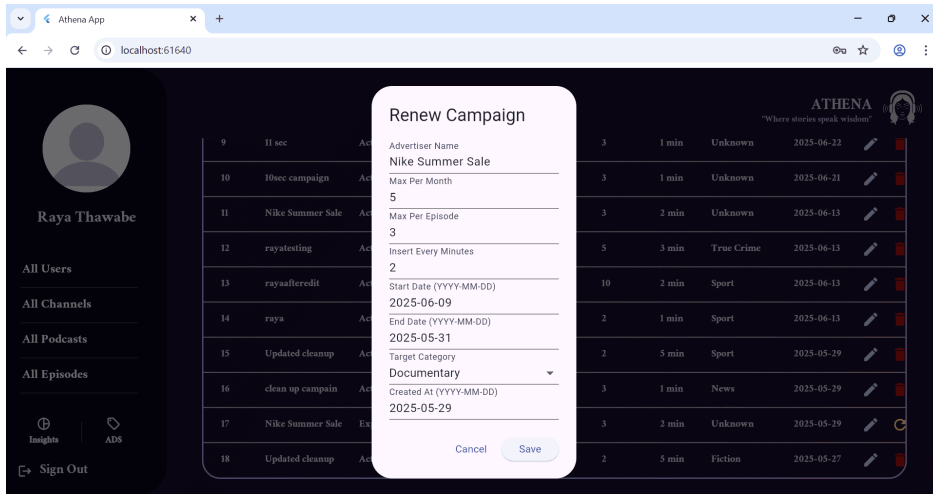


Figure 3.112: Renew Campaign - Ads Manager Part.5

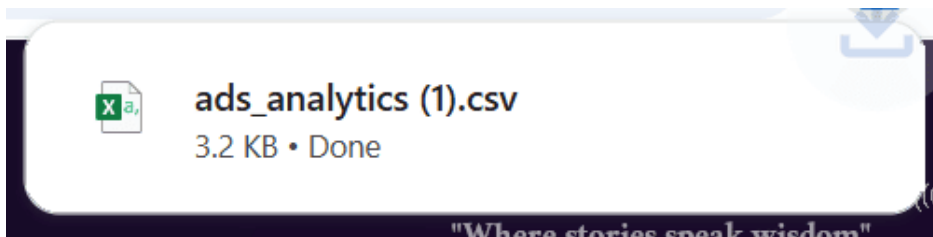


Figure 3.113: Download All Campaigns Data as a CSV - Ads Manager Part.6

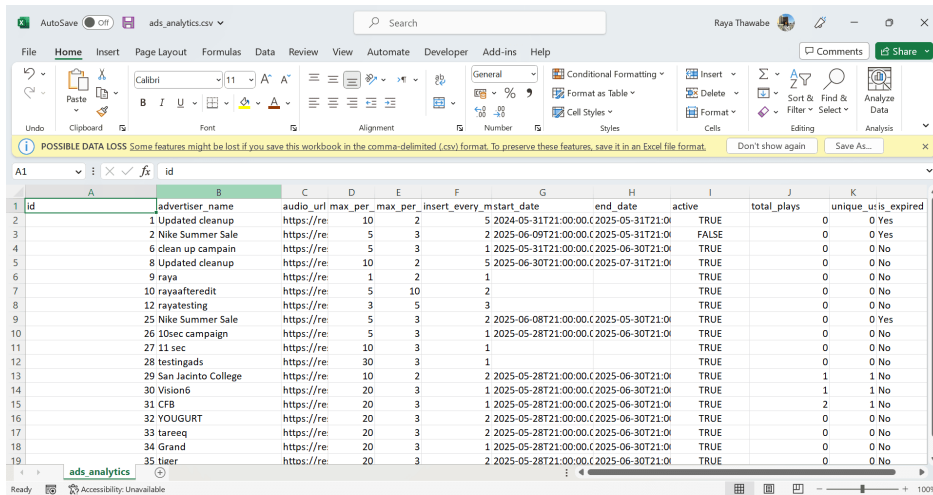


Figure 3.114: Example of CSV - Ads Manager Part.7

3.5 Standards and Specifications

Athena adheres to modern development standards and specifications:

- RESTful API design principles.
- JSON as data interchange format.
- JWT (JSON Web Token) for secure authentication.
- OpenAPI-style documentation for internal API reference.
- Accessibility support: Right-to-left (RTL) support for Arabic.

3.6 Constraints

3.6.1 Economy

To save on costs, Athena was developed on open-source tools and services. During development, the free tiers of Cloudinary and Whisper were used.

3.6.2 Environment

Custom power-efficient mobile design was considered, and the back-end infrastructure can be deployed to cloud platforms while keeping sustainability in mind.

3.6.3 Society

Athena supports both Arabic and English for inclusiveness and cultural accessibility. Transcriptions and AI summaries make it accessible to people with hearing disabilities.

3.6.4 Ethics and Privacy

User data is stored securely; passwords are hashed using bcrypt. Only the minimum amount of personal data is collected. All AI interactions are done locally or via vetted APIs.

4 Results

The Athena podcast architecture was developed with Node.js (Express) as the back-end and two microservices with FastAPI as the basis for audio matching and recommendations. Test data in the form of real podcasts and episodes, simulated user events was used to test the system.

User registration and log-in were made secure with JWT authentication, email verification, and password recovery through 6-digit codes. The recommender system used TF-IDF and cosine similarity to provide user-specific recommendations based on the user mood, interest, and history. It was tested to provide contextually relevant recommendations with low latency, thus providing a smooth user experience. OpenAI Whisper automatically transcribed the audio from the episode. Plain text and timestamped JSON transcripts were generated. For long audio files, transcripts were autochunked and joined together by the system. Transcriptions were readable and displayable, searchable, and highlightable in real life.

The review moderation module was able to effectively censor out offending content by combining blacklisting keywords and moderation API provided by OpenAI. All offending test submissions that were offensive or injurious in nature were detected, but clean reviews were handled seamlessly.

Real-time key activity such as episode upload, deletion, and admin approval prompted in-app notifications, with old read notifications periodically cleaned up using a scheduled task.

The ad management system provided admin to create and establish campaigns, assign targeting rules, and track ad plays. Audio ads were played between episodes without disturbing the main content. Admins could export campaign details in CSV format for in-depth analysis.

Generally, the system was stable, responsive, and well-integrated module-wise. All modules worked steadily under practical test scenarios.

5 Discussion

The goal of this project was to design an intelligent, scalable podcast platform with capabilities such as audio playback, recommendations, transcription, audio recognition, moderation, and ad management. The result shows that most of the basic features were developed and tested out against actual data.

The system now allows users to discover content based on their mood, interests, and listening history. Whisper-based transcription made it possible to display scripts and search within them. The audio matcher worked well for identifying episodes from recorded clips. Admin tool such as ad campaign creation, also review moderation, and notification management also worked reliably.

One of the strongest points of this project is its dependence on AI and automation—recommendations, transcription, moderation, and matching all rely on smart systems. One of the weak points is that there is no formal benchmarking and some reliance on third-party APIs, which could affect future scalability or cost.

Overall, the project was able to accomplish its main goals and have an operational prototype of a modern podcast platform. Additional user testing with actual users and even more advanced machine learning algorithms (like collaborative filtering) could potentially make the recommendations better. Additional features like real-time analytics or targeted advertising could also be explored.

6 Conclusions & Recommendation

6.1 Conclusions

This project has achieved to develop Athena, a complete podcast platform with intelligent recommendation, transcription, moderation, and audio recognition engines. The platform enables users to find and interact with podcast content rapidly through mood-based suggestions, search-based transcripts, in-app notifications, and admin-curated campaigns.

The results indicate that an infrastructure powered by AI with lightweight weights can provide valuable user experiences with non-intensive resource requirements. In-app notifications, Whisper transcription, and TF-IDF recommendation worked well for the first version of the app.

One of the main lessons learned from the project is that the incorporation of modular back-end services and valuable AI tools can provide engaging features even with limited volumes of user data. The design choices were driven by scalability, modularity, and user relevance.

6.2 Recommendation

like the formal evaluation of system performance through benchmarks and logs, and more personalized based on behavior monitoring.

Future work may include:

- Enhancing recommendations by training machine learning models once a large volume of user interaction data is collected.
- Migration to collaborative filtering models when there is more interaction data.
- Providing premium features such as ad skipping, access to the full transcript, and campaign promotion tools.
- Allowing user-curated lists of episodes, friend sharing, and role-based admin consoles.
- Combining real-time learning and personalization with implicit feedback like skip behavior and listen time.

The platform has a good foundation to work on in creating an scalable podcast platform. The more user interaction is witnessed, the more sophisticated the AI components can get to provide deeper personalization and smarter content discovery.

References

- Backlinko. (2025). *Podcast statistics you need to know*. Retrieved March 30, 2025, from <https://backlinko.com/podcast-stats>
- DemandSage. (2025). *Podcast statistics 2024 – listeners, revenue, growth trends*. Retrieved June 24, 2025, from <https://www.demandsage.com/podcast-statistics/>
- Northwestern University in Qatar. (2022). *Podcast listening in the middle east — mideast media survey 2022*. Retrieved March 24, 2025, from <https://www.mideastmedia.org/survey/2022/chapter/music-and-podcast/>
- Podcast Inc. (2024). *Podcast industry in the arab world: Market trends and growth*. Retrieved March 24, 2025, from <https://www.podcastinc.io/post/podcast-industry-in-the-arab-world>
- Podcaststatistics Team. (2025). *Podcast listener statistics and growth in 2025*. Retrieved June 24, 2025, from <https://podcaststatistics.com/>
- Shaher, M. (2024). *The surge in popularity of podcasts in the middle east*. Retrieved March 24, 2025, from https://www.researchgate.net/publication/390580885_The_Surge_in_Popularity_of_Podcasts_in_the_Middle_East