audino: A Modern Annotation Tool for Audio and Speech

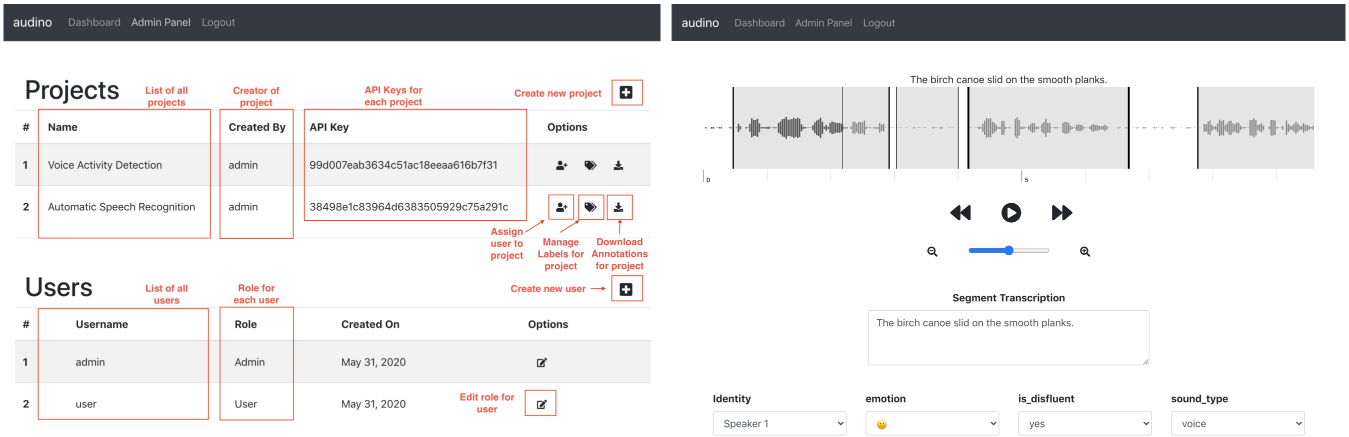
Manraj Singh Grover
Indraprastha Institute of Information Technology
Delhi, India
manrajg@iiitd.ac.in

Pakhi Bamdev
Indraprastha Institute of Information Technology
Delhi, India
pakhii@iiitd.ac.in

Yaman Kumar
Indraprastha Institute of Information Technology
Delhi, India
yamank@iiitd.ac.in

Mika Hama
Second Language Testing Inc.
Princeton, United States
mika.hama@2lti.com

Rajiv Ratn Shah
Indraprastha Institute of Information Technology
Delhi, India
rajivratn@iiitd.ac.in



Projects

#	Name	Created By	API Key	Options
1	Voice Activity Detection	admin	99d007eab3634c51ac18eeaa616b7f31	
2	Automatic Speech Recognition	admin	38498e1c83964d6383505929c75a291c	

Users

#	Username	Role	Created On	Options
1	admin	Admin	May 31, 2020	
2	user	User	May 31, 2020	

The birch canoe slid on the smooth planks.

Segment Transcription

The birch canoe slid on the smooth planks.

Identity	emotion	is_disfluent	sound_type
Speaker 1	😐	yes	voice

Figure 1: Sample screenshots of audino user interface rendered by a web browser. [left] Admin panel with marked regions explaining various functionalities available and accessible to users with admin roles. [right] Annotation panel with sample audio being annotated by the assigned user.

ABSTRACT

In this paper, we introduce a collaborative and modern annotation tool for audio and speech: audino. The tool allows annotators to define and describe temporal segmentation in audios. These segments can be labelled and transcribed easily using a dynamically generated form. An admin can centrally control user roles and project assignment through the admin dashboard. The dashboard also enables describing labels and their values. The annotations can easily be exported in JSON format for further processing. The tool allows audio data to be uploaded and assigned to a user through a key-based API. The flexibility available in the annotation tool enables annotation for Speech Scoring, Voice Activity Detection (VAD), Speaker Diarisation, Speaker Identification, Speech Recognition, Emotion Recognition tasks and more. The MIT open source license allows it to be used for academic and commercial projects.

CCS CONCEPTS

• **Applied computing** → **Annotation**; **Annotation**.

KEYWORDS

audio annotation, labelling, open source software, speech grading, voice activity detection, speech recognition, speaker diarisation, emotion recognition

ACM Reference Format:
Manraj Singh Grover, Pakhi Bamdev, Yaman Kumar, Mika Hama, and Rajiv Ratn Shah. 2020. audino: A Modern Annotation Tool for Audio and Speech. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Over the past few years, there has been a dramatic improvement in audio and speech research. The rise and performance of deep

Woodstock '18, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

neural network have achieved state-of-the-art results on various speech and audio tasks [5, 7, 11, 17]. These networks are necessary to consume and discover information in large volumes of data being published on the web. This necessitates the need to annotate data efficiently at scale for supervised learning of network. In this paper, we present a flexible and modern web-based annotation tool for audio and speech data called audino. The tool aims to provide a broad set of features required for annotation of speech datasets while focusing on increasing collaboration, project management and accessibility. The annotation tool is permissively licensed MIT[1] allowing it to be freely used for both academic research as well as commercial use. audino can be downloaded from `https://github.com/midas-research/audino`.

Many annotation tools already exist for image [14, 16], text [4, 13, 15] and speech [2, 6, 9] modality, where most of them require software installation on annotator's system. Recently, there has been increased interest in developing web-based annotation tools [3, 10, 13, 15]. Moving annotation tools to the web offer several advantages including data security, management and accessibility. A large number of these tools allow loading data, processing and saving annotations on annotator's web browser, while others offer server-side data loading and annotation storage. For speech modality, however, none of the annotation tools to our knowledge offer advantages of a server-side annotation tool. With this motivation, we developed audino.

We share and discuss the salient features of the tool below:

- **Accessibility.** In contrast to most annotation tools which need to be installed and run on annotator's system, audino is a web-based tool which makes it much easier to access through a web browser remotely. The side-effect of having data on annotator's system and the need to load a new datapoint after every datapoint annotation completion is mitigated.
- **Centralized control of data allocation, project management, and annotations.** In contrast to offline tools available, audino secures data access and simplifies project management through centralization. All labels are controlled centrally, which makes it less prone to error. The annotations are saved in a central database, making it easier to consume.
- **Easy setup and deployment.** The project makes use of Docker [12] to deliver the software easing the setup process, deployment and also scaling of the tool.
- **Security.** The application implements JSON Web Token [8] based authentication and authorization for secure login. An annotator can only view projects they are part of and can only access datapoints assigned to them. The audio filenames of all data points are hashed to prevent remote scraping further increases data security.
- **Multi-language and emoji support.** The tool supports Unicode character set, which enables annotation of multi-language datasets for tasks like Code-Switched [1] Automated Speech Recognition.

The paper is organized as follows. Section 2 describes the software design of audino. Section 3 explains the workflow of the application from the perspective of the end-user. Section 4 elaborates on various functionalities available on the main annotation dashboard. Finally, Section 5 summarizes the tool and discusses the future roadmap of the application.

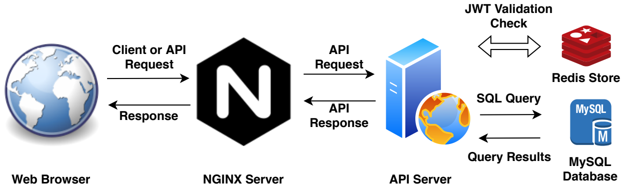


Figure 2: High level architecture of audino.

2 SOFTWARE DESIGN

audino is a production-ready web application tool. Figure 2 provides a high-level overview of the working of different components in the tool. Its client-side is platform-independent and can run on any modern browser. The server side serves the REST API and static content. All annotations and application data are stored on the server. We describe the software design in detail in the following sections.

2.1 Data Storage

The tool requires three types of data to be stored:

(1) Application data
(2) User session
(3) Audio data

Application data includes users, roles, projects, data, labels, and annotations generated. This data is stored in a structured format in a dockerized SQL database. The entity-relationship diagram for the database is shared in the Github public repository. For the current version, the tool supports MySQL[2] database; however, it can easily be extended to other SQL databases available.

To store the current user session, the application uses a dockerized Redis[3] store. It is an in-memory NoSQL store for fast retrieval of data. The application generates JSON Web Token ID for every user login and saves it in the store (with an expiration time) for future authentication.

The audio data uploaded is saved at a defined path inside the backend docker container. The application generates a unique filename for each uploaded file and stores the name inside the SQL database. The application then serves this file on request. The tool currently supports WAV, MP3 and OGG file formats as all browsers widely support these.

All of the containers use docker volume to persist the data. It is, however, advisable for users to mount a volume and map it to respective in-container paths to safeguard the data.

2.2 Server Side

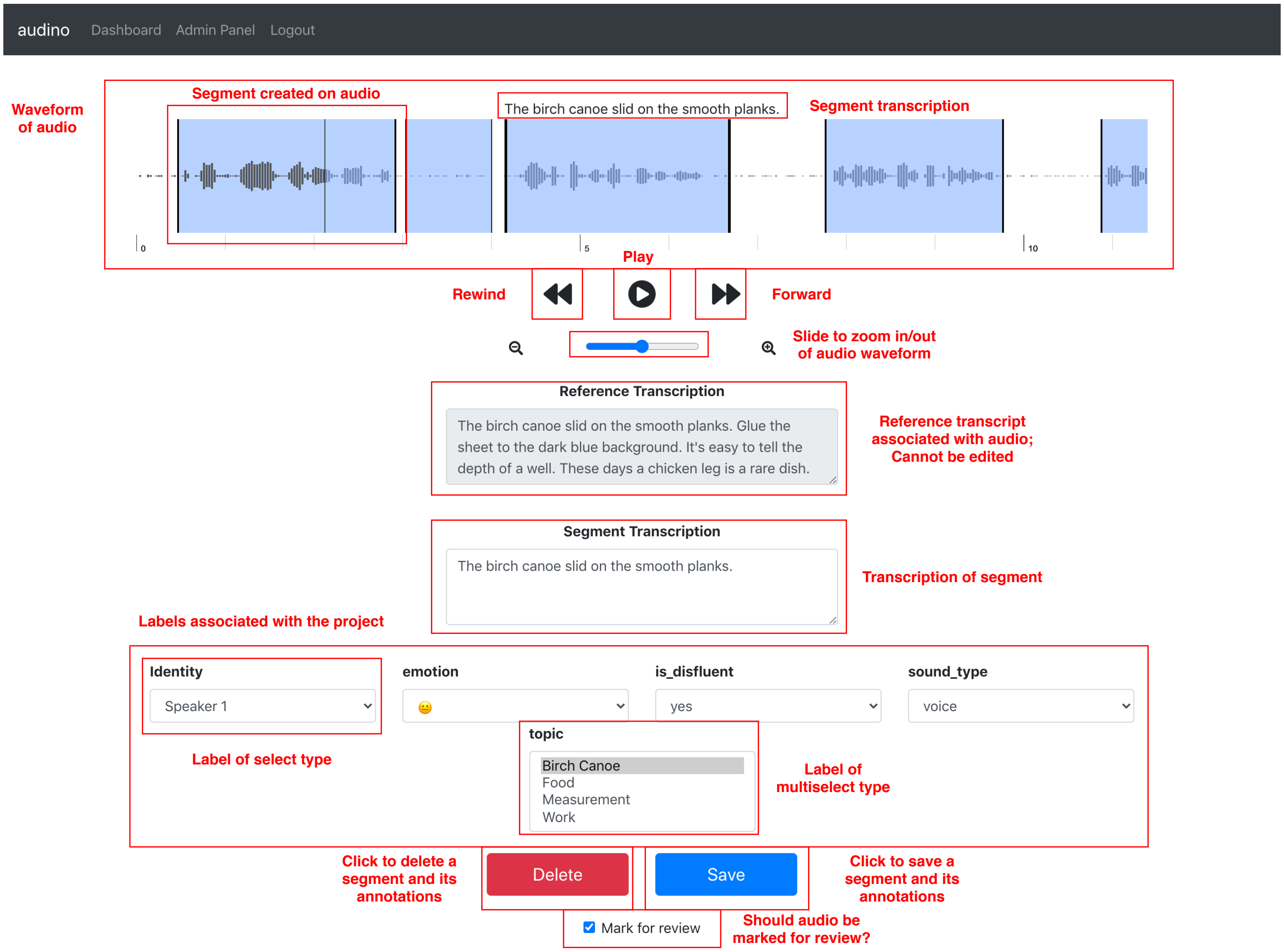
In addition to storage components discussed in the previous section, the server-side of the tool also includes an NGINX[4] server and an

[1] https://opensource.org/licenses/MIT

[2] https://www.mysql.com/

[3] https://redis.io/

[4] https://www.nginx.com/



Waveform of audio

Segment created on audio

The birch canoe slid on the smooth planks.

Segment transcription

Slide to zoom in/out of audio waveform

Reference Transcription

The birch canoe slid on the smooth planks. Glue the sheet to the dark blue background. It's easy to tell the depth of a well. These days a chicken leg is a rare dish.

Reference transcript associated with audio; Cannot be edited

Segment Transcription

The birch canoe slid on the smooth planks.

Transcription of segment

Labels associated with the project

Identity	emotion	is_disfluent	sound_type
Speaker 1	😐	yes	voice

topic

- Birch Canoe
- Food
- Measurement
- Work

Label of select type

Label of multiselect type

Click to delete a segment and its annotations

Click to save a segment and its annotations

Should audio be marked for review?

Figure 3: Screenshot of annotation dashboard showcasing various components.

API server. NGINX is a high-performance web server which can also be used for reverse proxy and HTTP cache. The application utilizes a dockerized NGINX server to serve static client-side content and a REST API using reverse proxy. The REST API server runs in a separate docker container using uWSGI[5]. The tool uses a Python-based framework called Flask[6], and its plugins to provide a RESTful API. This API allows authentication by checking the Redis store for request user's session. The API also enables the client-side to perform CRUD operations on the database. To interact and perform database operations, the API uses SQLAlchemy[7] library. Also, it provides a layer over database allowing easy switching to other SQL databases available. Alembic[8] library is used for versioning and migrating the database.

2.3 Client Side

The client-side interface is written mainly in HTML, CSS and JavaScript. The user interface is broken into individual components and developed using React[9], a JavaScript library for building user interfaces. React allows wiring of these client-side components with respective handlers as well as the REST API. Based on user interactions, the React components are rendered, and API requests are made. To make the application work for all screen size, the interface is styled using Bootstrap[10] CSS framework. The annotation dashboard leverages wavesurfer.js[11] library and its plugins for rendering audios and marking temporal regions. A production build is generated using React build system for NGINX to serve.

[5] https://uwsgi-docs.readthedocs.io/
[6] https://flask.palletsprojects.com/
[7] https://www.sqlalchemy.org/
[8] https://alembic.sqlalchemy.org/
[9] https://reactjs.org/
[10] https://getbootstrap.com/
[11] https://wavesurfer-js.org/

3 WORKFLOW

On opening the web application, a login screen is displayed. A user is required to have their account details in order to access the tool. An admin account is created during setup based on the information provided by the user. Account details of this user should be used to login for the first time. Once logged in, the user dashboard is displayed listing the projects assigned to the user. The user can click on a project name to move to a dashboard which lists audio datapoints assigned to that user for that project in a paginated manner. These datapoints are categorized based on their completion status and whether they are marked for review or not. On clicking on the filename of a datapoint, the annotation panel opens for that audio. We will describe the annotation panel in detail in Section 4.

The application also provides an admin panel accessible to users with admin role (illustrated in Figure 1 [left]). This panel allows admins to manage projects and users. An admin can create a new user, assign roles and projects to that user through this panel. The panel also allows the creation of new projects, labels and their associated label values, and download annotations for that project. For each new project, an API Key is generated, which allows uploading of new datapoints for that project.

4 ANNOTATION DASHBOARD

Figure 3 illustrates the annotation dashboard. The audio datapoint selected is rendered as a waveform. This component allows users to create temporal segments on audio for annotation. An audio control panel is provided to pause/play, move forward and backwards on the audio timeline. A zoom slider is also provided to control and zoom into a particular audio section for precise segmentation. The reference transcription is displayed below the control panel if provided when the datapoint was uploaded. On segment selection, a form consisting of segment transcript and associated project labels is displayed and is to be filled by the annotator. The annotator can save or delete any segment during the process, and the same will be reflected in the database. Finally, users can mark a datapoint for review. These datapoints are displayed under a separate category on the project's data dashboard.

5 SUMMARY AND ROADMAP

In this paper, we presented audino, a collaborative web-based modern annotation tool that allows temporal segmentation, transcription and labelling of language and speech aspects. We provide comprehensive documentation and tutorials to get the users started. The project has been under active development for a year now and has been used successfully for large-scale projects at our lab. Open sourcing the tool allows us to discover new possibilities of its utilization while enabling collaboration and easier management of dataset generation task.

The short-term roadmap of the project includes adding enhancements like user settings, feature-level permissions, and an analytics dashboard, which can offer insights into the quality of annotations generated, their statistics and agreement between annotators. The long-term roadmap includes improving test coverage of the project, adding continuous integration and delivery to development flow, adding project templates for speech-related tasks enabling more straightforward project setup, and leveraging recent state-of-the-art models for automatic labelling and transcription of audios (reducing overall annotation effort). We welcome everyone to contribute to the project and provide constructive feedback.

ACKNOWLEDGMENTS

We want to thank Anurag Agarwal, Karmanya Aggarwal and Hitkul Jangra for insightful discussions and suggestions, and reviewers for their comments.

REFERENCES

[1] Peter Auer. 2013. *Code-switching in conversation: Language, interaction and identity.* Routledge.

[2] Claude Barras, Edouard Geoffrois, Zhibiao Wu, and Mark Liberman. 2001. Transcriber: Development and use of a tool for assisting speech corpora production. *Speech Communication* 33, 1 (2001), 5 – 22. https://doi.org/10.1016/S0167-6393(00)00067-4 Speech Annotation and Corpus Tools.

[3] Abhishek Dutta and Andrew Zisserman. 2019. The VIA Annotation Software for Images, Audio and Video. In *Proceedings of the 27th ACM International Conference on Multimedia* (Nice, France) *(MM âĂŹ19)*. Association for Computing Machinery, New York, NY, USA, 2276âĂŞ2279. https://doi.org/10.1145/3343031.3350535

[4] Richard Eckart de Castilho, Éva Mújdricza-Maydt, Seid Muhie Yimam, Silvana Hartmann, Iryna Gurevych, Anette Frank, and Chris Biemann. 2016. A Web-based Tool for the Integrated Annotation of Semantic and Syntactic Structures. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*. The COLING 2016 Organizing Committee, Osaka, Japan, 76–84. https://www.aclweb.org/anthology/W16-4011

[5] Yusuke Fujita, Naoyuki Kanda, Shota Horiguchi, Kenji Nagamatsu, and Shinji Watanabe. 2019. End-to-End Neural Speaker Diarization with Permutation-free Objectives. In *Interspeech*. 4300–4304.

[6] Meghan Lammie Glenn, Stephanie M Strassel, and Haejoong Lee. 2009. XTrans: A speech annotation and transcription tool. In *Tenth Annual Conference of the International Speech Communication Association*.

[7] Manraj Singh Grover, Yaman Kumar, Sumit Sarin, Payman Vafaee, Mika Hama, and Rajiv Ratn Shah. 2020. Multi-modal Automated Speech Scoring using Attention Fusion. arXiv:2005.08182 [cs.CL]

[8] Michael Jones, John Bradley, and Nat Sakimura. 2015. *JSON Web Token (JWT)*. Technical Report. https://doi.org/10.17487/rfc7519

[9] Michael Kipp. 2001. Anvil-a generic annotation tool for multimodal dialogue. In *Seventh European Conference on Speech Communication and Technology*.

[10] Golan Levy, Raquel Sitman, Ido Amir, Eduard Golshtein, Ran Mochary, Eilon Reshef, Roi Reichart, and Omri Allouche. 2019. GECKO âĂŤ A Tool for Effective Annotation of Human Conversations. In *Proc. Interspeech 2019*. 3677–3678.

[11] Jason Li, Vitaly Lavrukhin, Boris Ginsburg, Ryan Leary, Oleksii Kuchaiev, Jonathan M. Cohen, Huyen Nguyen, and Ravi Teja Gadde. 2019. Jasper: An End-to-End Convolutional Neural Acoustic Model. In *Proc. Interspeech 2019*. 71–75. https://doi.org/10.21437/Interspeech.2019-1819

[12] Dirk Merkel. 2014. Docker: Lightweight Linux Containers for Consistent Development and Deployment. *Linux J.* 2014, 239, Article 2 (March 2014), 1 pages.

[13] Hiroki Nakayama, Takahiro Kubo, Junya Kamura, Yasufumi Taniguchi, and Xu Liang. 2018. doccano: Text Annotation Tool for Human. https://github.com/doccano/doccano Software available from https://github.com/doccano/doccano.

[14] Matthieu Pizenberg, Axel Carlier, Emmanuel Faure, and Vincent Charvillat. 2018. Web-Based Configurable Image Annotations. In *Proceedings of the 26th ACM International Conference on Multimedia* (Seoul, Republic of Korea) *(MM âĂŹ18)*. Association for Computing Machinery, New York, NY, USA, 1368âĂŞ1371. https://doi.org/10.1145/3240508.3243656

[15] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. brat: a Web-based Tool for NLP-Assisted Text Annotation. In *Proceedings of the Demonstrations Session at EACL 2012*. Association for Computational Linguistics, Avignon, France.

[16] Kentaro Wada. 2016. labelme: Image Polygonal Annotation with Python. https://github.com/wkentaro/labelme.

[17] S. Yoon, S. Byun, and K. Jung. 2018. Multimodal Speech Emotion Recognition Using Audio and Text. In *2018 IEEE Spoken Language Technology Workshop (SLT)*. 112–118.