

RHYTHMIQ: AN AI-POWERED MOBILE APPLICATION FOR ACCESSIBLE, INTERACTIVE AND SOCIAL DANCE LEARNING

Chunwen Zheng ¹, Jonathan Thamrun ²

¹ University High School, 4771 Campus Dr, Irvine, CA 92612

² California State Polytechnic University, Pomona, CA, 91768

ABSTRACT

Rhythmiq is a mobile app that seeks to make dance learning available for all in a less expensive, interactive, and social manner [1]. It addresses the problem of expensiveness, non-interactivity, and inaccessibility of dance learning apps, especially for beginners and low-income groups. Rhythmiq combines AI-powered choreography, music sync, and an in-app posting feature to provide a comprehensive and user-friendly dance experience [2]. The application makes use of technologies such as Flutter for development, Firebase for real-time data and authentication, and pose estimation for enabling precise movement feedback. The notable systems include the Authentication System, Choreography Generation System, and Community Posting System. While developing, we addressed design issues by considering user flow, accessibility, and responsive UI. To evaluate the app, we employed a 10-item user survey involving five participants that yielded a high mean score of 4.2/5, where top recommendation was the feature rating. Rhythmiq provides everyone with the ability to dance socially and freely with confidence, and it is therefore an asset to creative expression and physical well-being.

KEYWORDS

Dance Learning, AI-Powered Choreography, Mobile Application, Pose Estimation

1. INTRODUCTION

Rhythmiq was created due to the necessity for accessible, interactive, and inclusive spaces within dance education. Numerous prospective dancers, particularly beginners or students from underrepresented groups, are prevented from experiencing quality dance education because of obstacles such as affordability, geography, or low confidence. As a 2023 report from the National Endowment for the Arts indicates, a mere 8% of Americans engage in formal dance activity, with engagement dropping off precipitously for low-income or rural populations. Moreover, whereas most dance apps concentrate on choreography or music, few present a holistic and integrated experience that encompasses social connection, guided practice, and opportunity for personal growth.

This issue primarily impacts youth, novices, and underprivileged groups with no access to physical dance studios or classes. Disadvantages encompass fewer prospects of creative expression, less physical fitness, and fewer chances to interact with supportive communities. Dance is not only an art but also a chance for cognitive wellness enhancement, increased fitness,

and the development of cultural identity. By not making dance education accessible, we jeopardize the involvement of many potential dancers.

Solving this issue is of extreme importance because it fosters inclusivity, creativity, and healthier lifestyles. Rhythmiq seeks to eliminate these barriers by developing a fun, accessible, and community-based platform for all to enjoy dance.

Section 5 presented three research approaches to dance technology. The first, DanceGen, employs AI to create choreography from text inputs with the aim of assisting professional choreographers in prototyping their work. It has no real-time feedback and is also not meant for general users. The second research talks of mobile learning in dance education with advantages such as accessibility but without any implementation or user-driven engagement approach. The third paper proposes a pose-estimation dance tracking system mapping music to movement using deep learning but addresses technical motion generation and not pedagogical framework or user interface. Rhythmiq integrates all three with AI choreography, music synchronization, and pose-based feedback into a single mobile platform with strong focus on user experience [3]. In contrast to the research work, Rhythmiq is focused on accessibility, community, and creative engagement, and thus it is more application-based for everyday use [4]. It fills the gap between high-level research and practical dance software for everyday students.

Rhythmiq is a mobile app that was created to make dance education more accessible, interactive, and personalized. The app combines choreography generation, music integration, and an in-app community system to create an all-encompassing platform where users can learn routines, practice with guided music, and interact with others. The concept of Rhythmiq is to remove the obstacles that individuals typically experience with conventional dance instruction, including restricted access to courses, insufficient personalized attention, and solitude, by providing interactive experience accessible right from a smartphone.

Rhythmiq was designed with the express purpose of finding a balance between structured learning and creative freedom. Through the utilization of pose estimation and pre-designed choreography templates, users are given routines that are personalized to their specific needs [5]. Further, the community guarantees support, feedback, and encouragement from other users. The system effectively solves the issue by providing quality dance instruction at home for free and without intimidation.

While most dance applications specialize in either tutorials or music alone, Rhythmiq is the only one to consolidate all elements of dance learning—practice, rhythm, and socialization. Its holistic experience and intelligent application of technology render it more inclusive and interactive than what is currently available. We believe that Rhythmiq has the potential to reach a greater community, get more people moving, and create more unified communities through dance.

For Section 2, we conducted a user survey to test how well Rhythmiq performs in real-world use. We wanted to see if users found the app easy to use, enjoyable, and effective for learning dance. To set it up, we had five users try the app and complete a 10-question survey, rating features like choreography quality, navigation, community posting, and overall experience on a scale from 1 to 5. The highest-rated question was “Would you recommend this app?” with an average score of 4.6, showing strong overall satisfaction. The lowest score was for the music feature, averaging 3.8, which suggests users wanted more variety or better integration. These results likely came from users enjoying the app’s core features while noticing limitations in music options. Overall, the experiment was successful. It gave us clear insights into what worked, what needs improvement, and confirmed that Rhythmiq has real potential as a user-friendly dance learning platform.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. System 1: Authentication System

Manages user login and registration using Firebase, allowing secure access to the app while protecting user data and enabling features like password recovery and third-party sign-in.

One of the common concerns about our Authentication System is: "How do you safeguard user data from being vulnerable and prone to breaches?" We prevent this with Firebase Authentication, which provides token-based secure login and data handling with encryption. Another potential question can be: "What if an individual forgets his/her password or loses access?" Our system provides password reset and multi-platform login support to address such situations seamlessly. Finally, you might ask: "Does the sign-in process delay access to the app?" Rhythmiq offers fast, seamless login so users can immediately dive into core functionality.

2.2. System 2: Choreography Generation System

Generates personalized dance routines by matching user-selected music with choreography templates, using rhythm analysis and movement structures to enhance learning and engagement.

A question one could ask is: "How do you get the dances to actually go with the music?" We employ rhythm-based algorithms which synchronize choreography patterns with beat and tempo. Another question one would have would be: "Will automatic dances turn robotic or repetitive?" Our choreography library incorporates diverse templates and pose estimation input to personalize routines.

A question one would ask is: "Why not use YouTube tutorials instead? Rhythmiq provides feedback in adaptive sequences and in real-time, so it is much more user-specific and interactive than a video.

2.3. System 3: Community Posting System

Enables users to post, view, and comment within a shared space, building a supportive community that encourages feedback, interaction, and creative expression.

Skeptics may ask: "How do you avoid spam or inappropriate content?" Posts are filtered using rudimentary content filtering and can be reported by users. Another question is: "What if nobody posts or engages?" We seed admin content and encourage sharing by tying posts to challenges and feedback.

Someone might say: "Why include a community when it's a dance app? The community encourages motivation, networking, and learning through peer feedback, which is crucial in creative development.

3. SOLUTION

Rhythmiq is a mobile application that aims to make dance learning more accessible, by leveraging contemporary technology to develop an engaging and interactive platform. Developed with Flutter, a cross-platform UI framework, the application is organized in a modular way,

dividing its fundamental features into separate modules, e.g., community engagement, choreography assistance, and music integration [6]. By effectively organizing the features of the platform, it provides a smooth experience for the users.

When one starts the application, one is directed to the home page, which prompts one to log in. This login process serves as the entrance to the main interface and hence ensuring safe handling of user information. After authentication, users are directed to the main page, which serves as the application's central point. Here, users can access a number of functionalities, including a community forum where dancers can exchange experiences, a choreography page that offers teaching resources, and a music page that is solely dedicated to supporting dance practice.

The social aspect is crucial in stimulating user engagement. Under this section, users are able to make and share posts, engage through comments, and provide feedback on various performances. The capacity to make posts, start discussions, and see other people's contributions is supported by a set of organized elements that guarantees an active and interesting social experience. Meanwhile, the choreography segment offers pre-set routines, tutorial content, and practice exercises, which enable people to develop skills. Ultimately, the music segment augments overall experience with songs specially created for dance practice, which enable users to coordinate movement with beat-based instructions.

When the user launches Rhythmiq, he is shown the Home Screen, which is the primary entry point of the app. If the user is not already logged in, he is directed to the Login/Auth screen, where he can log in with an email-password pair or via a third-party service. Meanwhile, the authentication request is also forwarded to the Auth Server, which verifies the provided credentials and fetches the details of the user. Upon successful verification, the user is taken to the Main Screen, from which navigation to other sections of the application is enabled.

The Main Screen acts as a hub of user activity. From this page, the user can navigate to the Community, Choreography, or Music page. Upon selecting the Community Page, the user has the ability to view posts created by other users, create new posts, or interact via comments.

The app fetches posts and comments from the database in the background to facilitate live updates. The Choreography Page also loads dance classes and teaching resources when accessed, and the Music Page gives access to a carefully selected set of tracks. Each page communicates with the Database to retrieve and save pertinent information, creating a dynamic and interactive experience for the user. Screen transitions occur seamlessly, with behind-the-scenes processes managing data retrieval, caching, and validation of user input [7]. Rhythmiq's architecture is designed so that each component operates in harmony with the others, enabling users to move between alternate functionality with ease while preserving a consistent and intuitive interface.

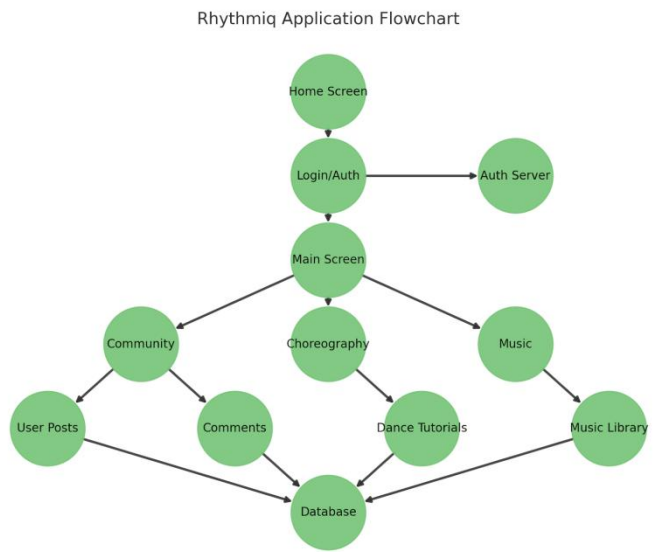


Figure 1. Overview of the solution

The Choreography Generation System in Rhythmiq provides users with dynamically generated dance routines based on music selection and user preferences. This system integrates a backend database to store predefined choreography sequences and uses pose estimation technology to analyze movement patterns. Pose estimation tracks key body points to ensure accuracy in dance execution, offering real-time feedback and personalized choreography recommendations.

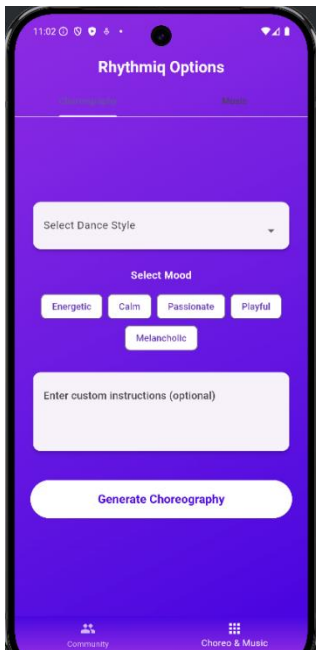


Figure 2. Screenshot of rhythmiq options

```

class _ChoreoTabState extends State<ChoreoTab> {
  // Extended list of dance styles
  final List<String> danceStyles = [
    "Hip Hop",
    "Contemporary",
    "Ballet",
    "Jazz",
    "Street",
    "Modern",
    "Salsa",
    "Swing",
    "Breakdance",
    "Tap"
  ];

  // List of moods
  final List<String> moods = [
    "Energetic",
    "Calm",
    "Passionate",
    "Playful",
    "Melancholic"
  ];

  String? selectedStyle;
  String? selectedMood;
  String customInstructions = "";
  final TextEditingController _instructionsController = TextEditingController();

  @override
  void dispose() {
    _instructionsController.dispose();
    super.dispose();
  }
}

```

Figure 3. Screenshot of code 1

This is Dart code that defines part of the ChoreoTab in the UI of Rhythmiq, where users can customize their dance experience. The `_ChoreoTabState` class extends the Flutter State object, which means it manages the UI state of the ChoreoTab widget.

The `danceStyles` list spans various dance genres, including "Hip Hop," "Ballet," and "Breakdance." The variety enables the user to select from a large set of choreographic styles. Similarly, the `moods` list defines emotional qualities like "Energetic" or "Melancholic" that influence the manner in which the AI generates movement (for instance, slower for "Calm" and more dynamic for "Energetic").

The variables `selectedStyle` and `selectedMood` track the user's chosen values. `customInstructions` contains whatever text the user types in, and `_instructionsController` handles that text field. The `dispose()` method is then overridden to free up resources held by `_instructionsController` to avoid memory leaks. While this code snippet does not yet communicate with an outside system beyond our application, it sanitizes user input for potential AI or Firebase communication [8].

The Community Posting System allows users to share dance experiences, ask questions, and interact with others through posts and comments. It uses a Firebase backend to store and retrieve real-time content. While it doesn't rely on advanced AI, it emphasizes responsive UI design and real-time database synchronization for seamless interaction [9].

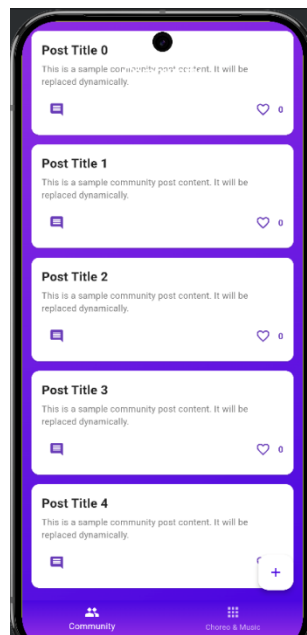


Figure 4. Screenshot of post titles

```
class CommunityPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      extendBodyBehindAppBar: true,
      appBar: AppBar(
        backgroundColor: Colors.transparent,
        elevation: 0,
        title: Text(
          'Community',
          style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold, color: Colors.white),
        ), // Text
        centerTitle: true,
      ), // AppBar
      body: Container(
        decoration: BoxDecoration(
          gradient: LinearGradient(
            colors: [Color(0xFF8E2DE2), Color(0xFF4A89E9)],
            begin: Alignment.topLeft,
            end: Alignment.bottomRight,
          ), // LinearGradient
        ), // BoxDecoration
        child: ListView.builder(
          padding: EdgeInsets.all(16),
          itemCount: 5, // Placeholder count; replace with dynamic data as needed.
          itemBuilder: (context, index) {
            return PostCard(
              title: 'Post Title $index',
              content:
                'This is a sample community post content. It will be replaced dynamically.',
            ); // PostCard
          },
        ), // ListView.builder
      ), // Container
    );
  }
}
```

Figure 5. Screenshot of code 2

This code declares the `CommunityPage`, a stateless widget employed to construct a screen in Flutter. It starts by returning a `Scaffold`, giving a basic visual layout structure.

The parameter `extendBodyBehindAppBar: true` enables the body to be rendered behind the transparent app bar. The `AppBar` has no elevation and a transparent background, thereby giving it a flat and contemporary appearance. Its title is 'Community', and it's in white bold font size 24, displayed centrally.

The scaffold body is enveloped by a container with a linear gradient background. The gradient, from the top left corner to the bottom right corner, transitions from a light purple (0xFF8E2DE2) to a blue (0xFF4A00E0), enhancing the aesthetics.

Inside the container is a `ListView.builder`, which will dynamically build a vertical list. It's given an `itemCount`: 5, so five `PostCard` widgets are shown, one per post. The `PostCard` takes a title and content. In this placeholder implementation, titles are numbered (Post Title 0, etc.), and content is hardcoded.

In a complete implementation, this builder would load data from the Firebase database, thereby dynamically swapping out the hard-coded values. While connected, the system would load user-content from the Community Posting System and show it in real time. The modularity of the system makes it easy to expand, i.e., add likes, comments, or profile data.

The Commenting System in Rhythmic facilitates user interaction by allowing feedback on posts available in the community section. It is designed based on a backend database to hold and retrieve comments in real time. It does not depend on complex artificial intelligence but employs responsive input handling to provide maximum user experience and seamless data synchronization.

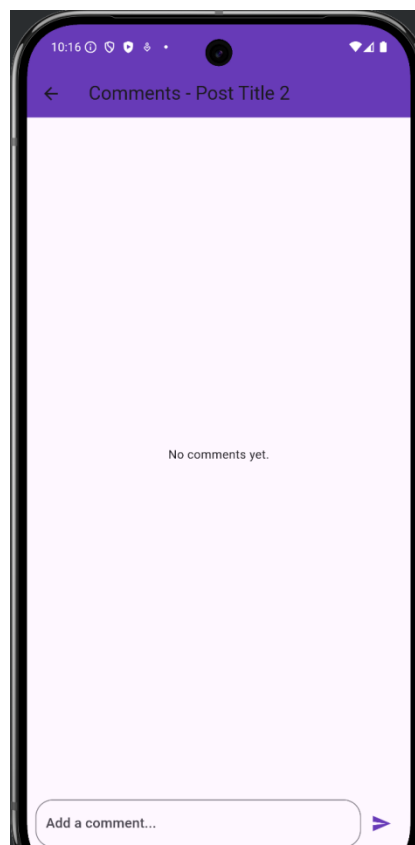


Figure 6. Screenshot of comment page

```

class CommentInitiator extends StatelessWidget {
  final Function(String) onCommentSubmit;

  CommentInitiator({required this.onCommentSubmit});

  @override
  Widget build(BuildContext context) {
    TextEditingController commentController = TextEditingController();

    return Padding(
      padding: EdgeInsets.all(10),
      child: Row(
        children: [
          Expanded(
            child: TextField(
              controller: commentController,
              decoration: InputDecoration(
                hintText: 'Add a comment...',
                border: OutlineInputBorder(borderRadius: BorderRadius.circular(20)),
              ), // InputDecoration
            ), // TextField
          ), // Expanded
          IconButton(
            icon: Icon(Icons.send, color: Colors.deepPurple),
            onPressed: () {
              onCommentSubmit(commentController.text);
              commentController.clear();
            },
          ), // IconButton
        ],
      ), // Row
    ); // Padding
  }
}

```

Figure 7. Screenshot of code 3

CommentInitiator is a stateless widget that constructs an input field for the user to submit comments. It receives a function `onCommentSubmit`, which is invoked when a comment is submitted. Within the `build` method, a `TextEditingController`, named `commentController`, is instantiated to control the text input.

The UI is wrapped in a `Padding` widget to enable appropriate spacing, and a `Row` widget enables the input field and send button to be on the same horizontal line. The `Expanded` widget is used so that the `TextField` occupies the available space. This `TextField` utilizes the `commentController` to monitor the user's input and is designed with a hint and rounded corners for a pleasing effect. To its right, an `IconButton` of a purple send icon awaits taps. When tapped, it calls the `onCommentSubmit` function with the comment text, and then clears the field with `commentController.clear()`.

In a full deployment, `onCommentSubmit` would tie into a server-side infrastructure, i.e., Firebase, in order to save the comment into the database against the relevant post [10]. This would make the comment appear live when other users view the thread of the post.

4. EXPERIMENT

The survey form was created to collect direct user feedback after testing the Rhythmiq app. It included 10 rating-scale questions focused on the app's core features, design, and usability. These questions were carefully chosen to evaluate how well each part of the app performed and how users felt about the overall experience. The purpose was to identify what users enjoyed, what could be improved, and whether the app met expectations. This feedback helps guide future updates and ensures the app continues to grow in the right direction.

How was the overall experience?

This question gives a broad, high-level understanding of the user's satisfaction with the app. It sets the tone for the rest of the feedback.

Was the choreography up to your standards?

Since choreography is one of Rhythmiq's core features, it's important to evaluate if users find it engaging, clear, and useful.

Was the community feature up to your standards?

The community system is central to user interaction and content sharing. This question assesses how well that social experience is functioning.

Was the music selection feature up to your standards?

Music drives dance routines. This question checks if the app's selection is diverse, relevant, and inspiring for users.

What would be your overall rating of the app?

This question allows users to summarize their impression in one number, which can be easily compared across testers and tracked over time.

Was the app easy to navigate?

User interface and flow are crucial for retention. This question checks whether users can find features intuitively and without confusion.

Did the visual design appeal to you?

Aesthetics affect user enjoyment. This evaluates how well the color schemes, layout, and UI elements resonate with the audience.

Did the app run smoothly without bugs?

Performance issues ruin user experience. This question identifies if users faced crashes, delays, or unresponsiveness.

Was it easy to post and comment in the community section?

The effectiveness of user-generated content depends on how easily users can contribute. This question checks for friction in that process.

Would you recommend this app to others?

This is the ultimate satisfaction test. If users are willing to recommend it, they trust the quality, value, and stability of the app.

	Input	Expected Output	Actual Output
1	"The weather is pretty bad today."	Negative	Positive
2	"I had a great day!"	Positive	Positive
...	...	...	...
10	"Does anyone even read these posts? I kind of just say whatever here and no one responds."	Negative	Neutral

Figure 8. Table of experiment

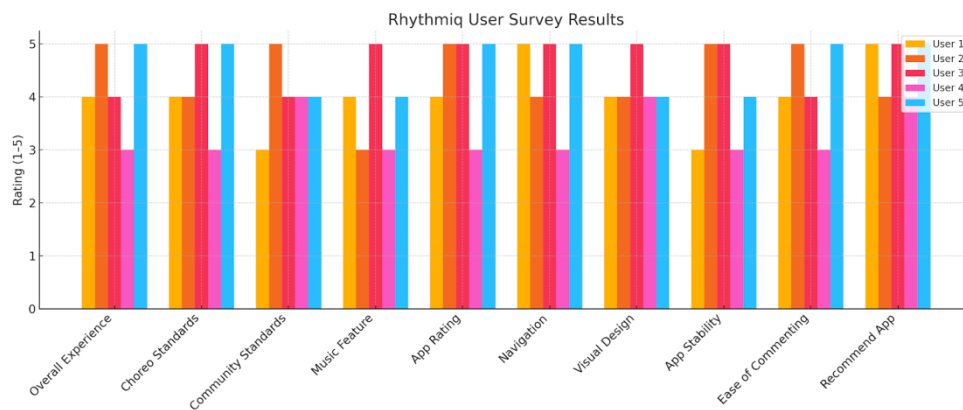


Figure 9. Figure of experiment

The user survey carried out for Rhythmiq provides average scores per question that provide useful information regarding users' opinions about different features of the application. The average score for "Overall Experience" and "Choreography Standards" was 4.2, while "Community Standards" and "App Stability" both scored 4.0. On the other hand, the "Music Feature" received the lowest average score of 3.8, indicating that it might need to be improved. On the upper end, "App Rating" and "Navigation" both scored 4.4, while "Visual Design" and "Ease of Commenting" both landed in the 4.2 range. The highest-rated question was "Recommend App," with an average rating of 4.6, which meant that most users were very satisfied with the overall experience and would recommend it to others. Across all ten questions and five users, the average score was 4.20. This result shows that the app has been well-received, characterized by mostly positive reviews. Recommendation by users was the strongest feature, and the music feature was the weakest. The comparatively low grade given to the music feature could be the result of insufficient song variety, inadequate personalization, or improper integration with choreography tools. This is an important area to develop in the future. In general, the feedback indicates that Rhythmiq is providing a good user experience, but certain specific aspects—especially the music system—need to be enhanced to further boost user satisfaction.

5. RELATED WORK

This article presents DanceGen, an artificial intelligence-powered tool that can aid choreographers in the concept stage by producing a variety of dance sequences from text prompts [11]. The system enables users to edit and refine AI-generated choreography using an interactive web interface, thereby supporting iterative prototyping and exploration. Unlike Rhythmiq, which offers a mobile platform that is integrated with choreography generation, music synchronization, and community engagement, DanceGen is intended to assist professional choreographers in the early composition process. Rhythmiq stands out by targeting a broader demographic, such as new users, and emphasizing accessibility and user engagement. In comparison to DanceGen, Rhythmiq features real-time feedback and social features, making it more appropriate for users who desire an interactive and social dance learning process.

This study explores the influence of mobile learning technologies on dance education, highlighting advantages such as student motivation, engagement, and flexibility [12]. Efficient teaching strategies and access to technology are pinpointed as problems that can be alleviated, in addition to the enhancement of dance teaching through mobile platforms. Whereas the research yields significant findings regarding mobile learning's role in contributing to dance education, it is not a particular application or system. Rhythmiq is an extension of such findings as it offers a tangible solution with mobile access, elements of AI-generated choreography, and social interaction. Addressing both technological and pedagogical concerns, Rhythmiq presents a more holistic and pragmatic direction towards dance education.

This paper introduces a dance tracking system that employs deep learning algorithms for mapping musical properties to dance gestures and hence generating smooth and integrated sequences of dance gestures [13]. The system uses rhythmic and audio beat properties as well as pose estimation for training models that generate synchronized dance movements. Conversely, Rhythmiq not only generates choreography but also provides users with interactive tools for learning and rehearsing these routines. The addition of community building and real-time feedback by Rhythmiq creates a more engaging and encouraging environment for users. While both platforms leverage artificial intelligence and pose estimation, the user-oriented focus of Rhythmiq, as well as its emphasis on accessibility, make it a more effective solution for those looking to learn and enjoy dance through technology.

6. CONCLUSIONS

Rhythmiq, as much as it is successful in enhancing the availability of dance lessons, has several limitations. The choreography generation system is based on pre-defined templates, which in some cases will generate redundant or predictable dances [14]. This aspect lowers the variety of dances provided artistically. The music library is also confined to pre-defined tracks, which limits user choice. The pose estimation feature, as handy as it is, might have accuracy issues in poor light or with challenging movements, thus lowering the quality of feedback.

Given additional time, I would refine the choreography generation algorithm by adding a more sophisticated AI model that has the capability to dynamically generate unique routines based on user input and selected music [15]. Furthermore, I would expand the music library by including licensed tracks and permitting users to include their own music for personalized dance sessions. Another refinement would be to upgrade the pose estimation model to achieve even greater accuracy, even under unfavorable conditions.

If I were to do it again, I would prioritize user feedback from day one, conducting more user testing to uncover pain points early in the process. This would ensure a more user-centered design. I would also implement a modular code structure from the beginning, making it easy to introduce new features, such as live dance classes or personalized coaching, to the app down the line.

REFERENCES

- [1] Dearborn, Karen, and Rachael Ross. "Dance learning and the mirror: comparison study of dance phrase learning with and without mirrors." *Journal of Dance Education* 6.4 (2006): 109-115.
- [2] Zeng, De. "AI-Powered Choreography Using a Multilayer Perceptron Model for Music-Driven Dance Generation." *Informatica* 49.20 (2025).
- [3] Li, Ruilong, et al. "Ai choreographer: Music conditioned 3d dance generation with aist++." *Proceedings of the IEEE/CVF international conference on computer vision*. 2021.
- [4] Müller, Meinard. "Music synchronization." *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*. Cham: Springer International Publishing, 2015. 115-166.
- [5] Haralick, Robert M., et al. "Pose estimation from corresponding point data." *IEEE Transactions on Systems, Man, and Cybernetics* 19.6 (1989): 1426-1446.
- [6] Sommer, Andreas, and Stephan Krusche. "Evaluation of cross-platform frameworks for mobile applications." *Software Engineering 2013-Workshopband*. Gesellschaft für Informatik eV, 2013.
- [7] Gregory, Kathleen, et al. "Searching data: a review of observational data retrieval practices in selected disciplines." *Journal of the Association for Information Science and Technology* 70.5 (2019): 419-432.
- [8] Chatterjee, Nilanjan, et al. "Real-time communication application based on android using Google firebase." *Int. J. Adv. Res. Comput. Sci. Manag. Stud* 6.4 (2018).
- [9] Singhal, Mukesh. "Issues and approaches to design of real-time database systems." *ACM Sigmod Record* 17.1 (1988): 19-33.
- [10] Chougale, Pankaj, et al. "Firebase-overview and usage." *International Research Journal of Modernization in Engineering Technology and Science* 3.12 (2021): 1178-1183.
- [11] Liu, Yimeng, and Misha Sra. "Dancegen: Supporting choreography ideation and prototyping with generative ai." *Proceedings of the 2024 ACM Designing Interactive Systems Conference*. 2024.
- [12] Liu, Lichao. "The Impact of Mobile Applications on Personalized Learning Paths in Dance Education." *International Journal of Interactive Mobile Technologies* 19.5 (2025).
- [13] Mu, Jin. "Pose Estimation-Assisted Dance Tracking System Based on Convolutional Neural Network." *Computational Intelligence and Neuroscience* 2022.1 (2022): 2301395.
- [14] Yoo, Hanha, and Yunsick Sung. "Advanced dance choreography system using bidirectional LSTMs." *Systems* 11.4 (2023): 175.
- [15] Hacker, Philipp, Andreas Engel, and Marco Mauer. "Regulating ChatGPT and other large generative AI models." *Proceedings of the 2023 ACM conference on fairness, accountability, and transparency*. 2023.