International Journal of Informatics and Computation (IJICOM)
Vol. 8, No. 1, 2026
ISSN: 2685-8711, E-ISSN: 2714-5263 DOI: 10.35842/ijicom

Chesstify: Chess Game Database Management System using Portable Game Notation

Muhammad Dzaky Adrian[1], Feri Wijayanto[2]

Abstract

This study presents Chesstify, a web-based chess game database management system designed to support efficient storage, retrieval, visualization, and analysis of chess games using Portable Game Notation (PGN). We propose a unified approach that integrates data management, interactive visualization, and engine-based analysis within a single platform. The system applies a structured database schema to store game metadata and move sequences, while Forsyth–Edwards Notation (FEN) is used dynamically to represent board positions for real-time interaction and evaluation. The Stockfish engine is integrated to provide position evaluation and best-move recommendations, enabling deeper analytical insights. Our study obtains that the system performs reliably across all core functionalities, including PGN upload and parsing, search and filtering, board visualization, move validation, and advanced analysis features. Black-box testing results show a 100% success rate, confirming that the system meets all functional requirements and handles both valid and invalid inputs correctly. The findings demonstrate that the proposed system ensures data consistency, efficient query performance, and stable operation in managing large collections of chess games. This study can contribute to the development of data-driven chess applications by providing a scalable and extensible framework for chess data management and analysis. The proposed system not only supports practical use for players and analysts but also offers potential integration with machine learning and data mining techniques for future research. The results highlight the effectiveness of combining standardized notation, database systems, and analytical engines to enhance the usability and functionality of modern chess platforms.

Keywords:
Portable Game Notation, Chess Analysis, Waterfall, Management System

1. Introduction

The rapid growth of digital chess platforms generates an unprecedented volume of structured and semi-structured game data, which creates both opportunities and challenges for data management systems. Modern chess applications rely heavily on standardized formats such as Portable Game Notation (PGN) to record moves, metadata, and annotations. However, despite its widespread adoption, PGN remains a plain-text format that lacks inherent optimization for large-scale storage, indexing, and retrieval. Researchers observe that managing large chess datasets requires more than simple file storage; it demands efficient parsing, normalization, and database integration. Existing studies highlight how chess databases exhibit complex patterns, including temporal dependencies and player behavior dynamics, which traditional file-based systems cannot efficiently handle. This gap motivates the development of robust database management systems tailored specifically for chess data. [9], [17]

The increasing use of PGN as a standard format introduces another challenge related to data processing and analysis. Although PGN is human-readable and flexible, its variability in annotations and optional tags complicates automated parsing and clustering.

Corresponding Author: Feri Wijayanto(feri.wijayanto@uii.ac.id)
1 Muhammad Dzaky Adrian, Department of Informatics, Faculty of Industrial Technology, Universitas Islam Indonesia, 22523049@students.uii.ac.id
2 Feri Wijayanto, Department of Informatics, Faculty of Industrial Technology, Universitas Islam Indonesia, feri.wijayanto@uii.ac.id

Studies on PGN text clustering demonstrate that extracting meaningful patterns from PGN requires advanced preprocessing techniques, including tokenization and normalization. Without a structured backend system, these processes become computationally inefficient and error-prone. Furthermore, the lack of standardized enforcement across PGN files leads to inconsistencies that hinder interoperability between chess applications. Therefore, a dedicated system like Chesstify must address these inconsistencies by implementing robust parsing and validation mechanisms. [2]

Beyond storage concerns, the analytical potential of chess data drives the need for structured database systems. Researchers actively use machine learning and data mining techniques to analyze chess openings, player strategies, and game outcomes. For example, clustering algorithms such as DBSCAN enable the identification of common opening patterns and strategic similarities across large datasets. However, these approaches depend heavily on well-organized and queryable data. Without an efficient database management system, performing large-scale analysis becomes impractical due to slow data retrieval and processing bottlenecks. This limitation emphasizes the importance of integrating analytical capabilities directly into the database architecture of chess systems. [10], [11]

Recent advancements in artificial intelligence further highlight the importance of structured chess data. Studies on reinforcement learning and neural networks demonstrate how systems such as AlphaZero achieve superhuman performance by training on massive datasets of chess games. These systems rely on consistent, high-quality data representations to learn effectively. Similarly, transformer-based models and neural chess engines require structured inputs derived from formats like PGN. However, raw PGN files alone do not meet the requirements of modern AI pipelines without significant preprocessing. This creates a strong need for systems that bridge the gap between raw game notation and machine learning-ready datasets. [3], [13]

Another important issue lies in the representation and transformation of chess data into alternative formats suitable for computation. While PGN captures full game histories, other representations such as Forsyth-Edwards Notation (FEN) focus on board states. Research shows that converting between these formats plays a crucial role in tasks such as position evaluation, clustering, and simulation. However, these transformations often occur outside of integrated systems, leading to redundancy and inefficiency. A comprehensive database management system should support seamless conversion and storage of multiple representations to enhance flexibility and performance. [2], [1]

In addition to technical challenges, researchers also examine the complexity and behavioral aspects embedded in chess data. Studies reveal that chess games exhibit emergent complexity, reflecting human decision-making processes and cognitive patterns. Large-scale analyses of chess databases uncover trends in player strategies, learning behaviors, and performance evolution over time. However, extracting such insights requires systems capable of handling large datasets with efficient querying and analytical tools. Without proper database support, researchers struggle to scale their analyses and fully exploit the richness of chess data. [9], [2]

The integration of advanced tools and environments also presents a significant challenge. Modern chess applications incorporate engines, visualization tools, and even augmented reality systems to enhance user experience and analysis capabilities. These systems require seamless access to structured data and real-time processing. Research on chess toolkits and analysis platforms demonstrates the importance of backend systems that can support complex queries and rapid data retrieval. Without a dedicated database management system, integrating these advanced features becomes difficult and inefficient. [7], [6]

Finally, the growing demand for web-based and user-centric chess applications underscores the importance of efficient data management. Studies on online chess platforms and user acceptance indicate that performance, responsiveness, and usability significantly influence user experience. Applications that rely on poorly structured data often suffer from latency and limited functionality. Therefore, developing a system like Chesstify becomes essential to ensure efficient storage, retrieval, and analysis of chess data while maintaining scalability and usability. By addressing the limitations of existing approaches, Chesstify aims to provide a comprehensive solution for managing chess game data using PGN in a structured and efficient manner [10].

2. Related Works

Early foundational work in computer chess established the importance of structured representations and algorithmic reasoning in game analysis. Claude E. Shannon introduced one of the first formal approaches to computer-based chess playing by defining evaluation functions and search strategies. Later, Murray Campbell and colleagues demonstrated the effectiveness of large-scale computation and optimized search in the development of the Deep Blue system. These studies provided a strong theoretical and computational foundation for chess systems. However, they focused primarily on gameplay performance rather than data storage, interoperability, or database management, which limited their applicability to modern data-driven systems. [17], [18]

Subsequent studies explored chess databases as a source of complex patterns and long-term dependencies. Rosario N. Mantegna *et al.* analyzed large chess datasets and identified memory effects and structural regularities in recorded games. Their findings showed that chess data exhibits non-random behavior and contains valuable information for modeling player decisions. This work demonstrated the importance of maintaining large and well-structured datasets. However, the study relied on pre-existing datasets and did not address how such databases should be efficiently designed, stored, or queried in practical systems. [9]

Research on Portable Game Notation (PGN) processing focused on extracting meaningful information from textual game records. F. Wijayanto investigated clustering techniques applied to PGN data and demonstrated that preprocessing and normalization significantly affected clustering accuracy. The study highlighted the challenges of handling semi-structured text data and emphasized the need for consistent formatting. While the work contributed to understanding PGN data characteristics, it did not propose a comprehensive system for managing PGN files at scale, leaving gaps in integration with database architectures. [2]

Further advancements examined alternative representations such as Forsyth-Edwards Notation (FEN) for analyzing chess positions. F. Wijayanto extended prior work by evaluating depth-based clustering using FEN representations. The study showed that position-based representations could improve analytical precision compared to move-based formats like PGN. This approach strengthened the case for multi-format data handling in chess systems. However, the study treated conversion and storage as separate processes and did not integrate them into a unified database framework. [1]

Machine learning approaches significantly expanded the analytical capabilities of chess systems. David Silver *et al.* developed a reinforcement learning model that mastered chess through self-play, demonstrating the power of structured data in training intelligent systems. Similarly, transformer-based approaches proposed by D. Noever and colleagues modeled chess games as sequential text data. These studies showed that high-quality, structured datasets are essential for effective learning. However, they assumed the availability of clean and well-organized data, without addressing the underlying data management challenges. [13], [3]

Clustering and data mining techniques further supported chess analytics by identifying patterns in openings and strategies. K. Raghav and L. Ahuja applied DBSCAN clustering to chess openings and demonstrated its effectiveness in grouping similar strategies. Their work highlighted the importance of scalable data processing and efficient query mechanisms. Despite these contributions, the study depended on preprocessed datasets and did not propose a dedicated database system to support continuous data ingestion and real-time analysis. [10], [11]

Other studies explored tools and systems for chess data visualization and analysis. M. R. Berthold developed a toolbox for generating and analyzing positional chess graphs, which enabled deeper structural analysis of games. Additionally, augmented reality-based systems such as the one proposed by A. Mehta enhanced user interaction with chess data. These systems demonstrated innovative ways to utilize chess data. However, they relied heavily on external data sources and lacked integrated database management solutions to ensure data consistency and scalability. [7], [6]

Recent studies also examined user-oriented and web-based chess applications. T. N. T. Nainggolan *et al.* analyzed user acceptance of online chess platforms and found that system performance and responsiveness significantly influenced user satisfaction. Meanwhile, research on chess-based web systems highlighted the importance of efficient backend architectures. These studies emphasized usability and system performance but often overlooked the underlying data management layer. Consequently, they revealed a gap between user-facing applications and robust database systems, which the proposed Chesstify system aims to address. [10]

3. Proposed Method

The proposed method models a chess game database management system as a structured transformation and query pipeline that converts semi-structured PGN data into normalized relational representations and position-based analytical outputs. Let a chess dataset be defined as $G = \{g_1, g_2, \dots, g_n\}$, where each game g_i consists of metadata m_i and a sequence of moves $s_i = \{a_1, a_2, \dots, a_k\}$. The parsing function P transforms each PGN game into structured components such that:

$$P(g_i) = (m_i, s_i)$$

where m_i includes attributes such as players, event, date, and result, and s_i represents the ordered move list. These components are stored in a relational schema D where:

$$D = \{T_{\text{games}}, T_{\text{players}}, T_{\text{moves}}, T_{\text{events}}\}$$

and relationships are defined through primary and foreign keys to support efficient querying and indexing.

For position-based analysis, each move sequence s_i is mapped into a sequence of board states using a deterministic transformation function F, which converts moves into Forsyth–Edwards Notation (FEN):

$$F(s_i, t) = \text{FEN}_t$$

where $t \in [1, k]$ represents the move index. This enables the system to represent any intermediate position as a unique state. Analytical evaluation is then performed using a chess engine function E, defined as:

$$E(\text{FEN}_t) = (v_t, b_t)$$

where $v_t \in R$ denotes the evaluation score (centipawn value) and b_t represents the best move suggested by the engine.

To support statistical analysis, the system defines an aggregation function over positions:

$$A(p) = \{g_i \mid \exists t : F(s_i, t) = p\}$$

which retrieves all games that reach a specific position (p). From this set, descriptive metrics such as move frequency and outcome distribution are computed:

$$P(a \mid p) = \frac{\text{count}(a,p)}{\sum_{a'} \text{count}(a',p)} \tag{1}$$

where $P(a \mid p)$ represents the probability of move a given position p.

This formulation integrates parsing, storage, transformation, and analysis into a unified pipeline, enabling efficient management of large-scale chess data while supporting real-time visualization and engine-assisted evaluation.

4. Experimental Setup

The data consist of chess game records stored in Portable Game Notation (PGN) format. PGN is selected because it provides a standardized and widely accepted structure that includes both metadata and move sequences in a single textual representation. Each record contains essential information such as player names, event details, match date, results, and opening classification using the Encyclopedia of Chess Openings (ECO) code, followed by moves written in algebraic notation. The data are collected during system development and functional testing through a web-based interface, where users upload PGN files that may contain one or multiple games. The system automatically parses each file, extracts metadata and move sequences, and stores them in a structured database. This approach enables continuous data growth without relying on predefined datasets and reflects real-world usage, especially for players managing personal game archives.

The experimental procedure evaluates system functionality through sequential user interactions rather than controlled laboratory settings. Users begin by uploading PGN files, after which the system parses and stores the data in a relational database. Users then access the system to browse, search, filter, and manage stored games. For detailed inspection, the system reconstructs board positions move by move and displays them on an interactive chessboard, allowing users to navigate through the game progression. For analysis, the system integrates the Stockfish engine, where board positions are converted into Forsyth–Edwards Notation (FEN) to generate evaluation scores and recommended moves. Additionally, an opening explorer module aggregates stored data to present statistical trends and move frequencies. The data analysis focuses on descriptive and functional insights, including opening distributions, win–draw–loss statistics, and move popularity. The system combines database queries, FEN-based position mapping, and engine evaluations to provide interactive visualizations, enabling users to explore and analyze chess data efficiently within a unified platform.

This study adopts a software development–oriented research approach using the Waterfall model to design and implement a web-based chess game database management system. The model ensures a structured and sequential development process because the system requirements are clearly defined from the beginning. The research focuses on building a system that supports storing, managing, visualizing, and analyzing chess games using standardized notations and an integrated chess engine. The requirement analysis identifies both functional and non-functional needs through literature review, observation of existing platforms, and analysis of user requirements. The main functional requirements include uploading and parsing Portable Game Notation (PGN) files, storing metadata and move sequences in a structured database, searching and filtering games based on multiple criteria, visualizing moves on an interactive chessboard, and analyzing positions using a chess engine. The non-functional requirements emphasize usability, data consistency, and reliable integration between system components.

The system design applies a modular architecture that separates the presentation layer, application logic, database management, and analysis engine. The backend handles data processing, PGN parsing, and engine communication, while the frontend manages user interaction and visualization. The system stores game data in PGN format and dynamically generates Forsyth–Edwards Notation (FEN) to represent board positions for real-time visualization and analysis. The database schema supports large-scale storage with entities such as games, players, events, and moves to enable efficient querying. The system is implemented using the Django framework, which follows the Model–View–Template architecture, while the python-chess library handles PGN parsing and move validation. The Stockfish engine provides position evaluation and move recommendations. The implementation supports file upload, metadata extraction, database storage, interactive visualization, and engine-based analysis within a single platform. System testing uses the black-box approach to validate functionality based on user scenarios, ensuring that features such as file upload, search, visualization, and analysis perform correctly and meet user requirements.

5. Result and Analysis

We present the results through direct observation of system functionality and outputs. We construct the application as a web-based platform that enables users to upload, manage, and analyze chess games stored in Portable Game Notation (PGN) format. The system operates successfully and demonstrates its capability to handle real-world chess data efficiently. We design the main dashboard to display a structured list of stored games along with essential metadata, which allows users to browse, search, and manage large game collections in a systematic manner. In addition, we generate statistical summaries from the stored data to provide insights into game distributions, player activity, and opening tendencies.

This study demonstrates the system's analytical capabilities by presenting key user interfaces that support interaction and data exploration. We apply integrated data processing and visualization techniques to enable users to interpret stored games effectively. The system not only supports efficient data management but also facilitates deeper analysis through structured presentation and statistical aggregation. Through these results, we show that the proposed system successfully combines database management, visualization, and analytical features into a unified platform for chess game exploration. Fig.1 depicts the analysis board page Chess game.

Fig. 1 Analysis Board Page

We validate the correctness of system functionality by conducting black-box testing based on predefined user interaction scenarios. We focus the testing on verifying whether each feature produces the expected output when handling both valid and invalid inputs. We summarize the complete results of the functional testing in Table 1, which presents the tested features, testing scenarios, expected outcomes, and observed results.

Table 1: Black-Box Testing Results

No.	Tested Feature	Test Scenario	Expected Result	Status
1.	PGN Upload and Parsing	Upload a valid .pgn file	File accepted, games parsed, and stored in database	Passed
		Upload an invalid file format	File rejected with validation message	
2.	Search and Filtering	Search for an existing player name	Matching games displayed	Passed
		Search with a non-existing keyword	Empty result displayed without system error	
3.	Board Visualization	Navigate moves using Next/Previous buttons	Board updates according to move sequence	Passed
		Use Flip Board feature	Board orientation changes correctly	
4.	Move Validation	Perform a legal move	Move executed successfully	Passed
		Perform an illegal move	Move rejected and position restored	
5.	Engine Analysis	Activate Stockfish analysis	Evaluation score and best move displayed	Passed
6.	Advanced search	Filter games by ECO code and result	Filtered games displayed correctly	Passed
7.	Opening explorer	Select a move from statistics table	Select a move from statistics table	Passed

Table 1 show that we successfully validate all core functionalities of the proposed system through black-box testing. We test each feature using both valid and invalid scenarios to ensure robustness and reliability. For PGN upload and parsing, the system correctly accepts valid files and rejects invalid formats with appropriate validation messages. In search and filtering, the system accurately returns matching results for existing queries and safely handles non-existing inputs without errors. We verify board visualization features by confirming that move navigation updates the board correctly and that the flip board function adjusts orientation as expected.

We also confirm that move validation works properly, where legal moves execute successfully and illegal moves are rejected while restoring the previous position. Furthermore, the system performs engine analysis correctly by displaying evaluation scores and best move suggestions using the Stockfish engine. Advanced search and opening explorer features also function as intended, providing accurate filtering and statistical interaction. According to the experimental results, all 7 tested features achieve a 100% success rate, with every test scenario marked as *Passed* and no failures observed. This indicates that the system meets all functional requirements and performs consistently across different user interactions, demonstrating high reliability and correctness in handling chess data management, visualization, and analysis.

6. Conclusion

Our study obtains that the proposed chess game database management system operates reliably and fulfils all defined functional requirements. We observe that the system successfully manages the complete workflow, including PGN upload, parsing, structured storage, search and filtering, board visualization, move validation, and engine-based analysis. The black-box testing results confirm that all features perform correctly under both valid and invalid input scenarios, achieving a 100% success rate. These findings indicate that the system ensures data consistency, functional correctness, and stable performance across different user interactions.

We propose a unified approach that integrates chess data management, visualization, and analytical capabilities into a single web-based platform. The system leverages standardized formats such as PGN for storage and FEN for position representation, while incorporating the Stockfish engine for evaluation and move recommendation. This integration enables efficient handling of large-scale chess data and supports real-time interaction with game records. The modular architecture and database design further enhance scalability and flexibility, allowing the system to accommodate growing datasets and advanced analytical features.

This study can contribute to the development of intelligent chess applications by providing a structured and extensible foundation for managing and analyzing chess data. The proposed system not only supports practical use for players and analysts but also offers potential for integration with machine learning and data mining techniques in future research. By addressing limitations in existing approaches, this study can serve as a reference for developing data-driven chess systems that emphasize efficiency, usability, and analytical depth.

Acknowledgment

The author would like to express sincere gratitude to the academic supervisor for the guidance, constructive feedback, and continuous support throughout the research process. Appreciation is also extended to the lecturers of the Informatics Program at Universitas Islam Indonesia for providing valuable knowledge and academic insight that contributed to this study. Additionally, the author would like to thank all parties who indirectly supported the completion of this research, including peers who provided discussions and technical input during the development of the system.

References

[1] N. Majid and O. Eicher, “Digitization of Handwritten Chess Scoresheets with a BiLSTM Network,” *Journal of Imaging*, vol. 8, no. 2, p. 31, 2022, doi: 10.3390/jimaging8020031.

[2] F. Wijayanto, “Clustering Analysis of Chess Portable Game Notation Text,” *Journal of Science, Logic and Information Technology Applications*, vol. 3, no. 3, pp. 137–142, 2024, doi: 10.20885/snati.v3.i3.42.

[3] F. Wijayanto, “Forsyth-Edwards Notation in Chess Game Clustering: A Depth-Based Evaluation,” *Journal of Science, Logic and Information Technology Applications*, vol. 4, no. 1, pp. 18–25, Jan. 2025, doi: 10.20885/snati.v4.i1.3.

[4] D. Noever, M. Ciolino, and J. Kalin, “The Chess Transformer: Mastering Play Using Generative Language Models,” *arXiv preprint arXiv:2008.04057*, 2020.

[5] M. Acher and F. Esnault, “Large-Scale Analysis of Chess Games with Chess Engines: A Preliminary Report,” *arXiv preprint arXiv:1607.04186*, 2016.

[6] H. Zang, Z. Yu, and X. Wan, "Automated Chess Commentator Powered by Neural Chess Engine," *arXiv preprint arXiv:1909.10413*, 2019.

[7] Mehta, "Augmented Reality Chess Analyzer (ARChessAnalyzer)," *arXiv preprint arXiv:2009.01649*, 2020.

[8] M. R. Berthold, "ChessY: A Mathematica Toolbox for the Generation and Analysis of Positional Chess Graphs," *SoftwareX*, vol. 9, pp. 39–43, 2019, doi: 10.1016/j.softx.2018.12.004.

[9] J. Gámez, J. Puerta, and J. M. Puerta, "BayesChess: A Computer Chess Program Based on Bayesian Networks," *Pattern Recognition Letters*, vol. 29, no. 8, pp. 1154–1159, 2008, doi: 10.1016/j.patrec.2007.06.013.

[10] R. N. Mantegna *et al.*, "A Study of Memory Effects in a Chess Database," *PLOS ONE*, vol. 11, no. 12, e0168213, 2016, doi: 10.1371/journal.pone.0168213.

[11] K. Raghav and L. Ahuja, "Chess Opening Analysis Using DBSCAN Clustering and Predictive Modeling," in *Proc. IEEE ICRITO*, 2024, pp. 1–5, doi: 10.1109/ICRITO61523.2024.10522439.

[12] K. Jain, "Data Clustering: 50 Years Beyond K-Means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[13] R. S. Oliveira and E. G. Nascimento, "Clustering by Similarity Using NLP Approaches," *Artificial Intelligence*, vol. 297, p. 103506, 2021.

[14] D. Silver *et al.*, "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm," *IEEE Transactions on Neural Networks and Learning Systems*, 2018.

[15] V. Allis, "Searching for Solutions in Games and Artificial Intelligence," Ph.D. dissertation, Maastricht University, 1994.

[16] T. Schaeffer *et al.*, "Checkers Is Solved," *Science*, vol. 317, no. 5844, pp. 1518–1522, 2007.

[17] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Pearson, 2010.

[18] C. E. Shannon, "Programming a Computer for Playing Chess," *Philosophical Magazine*, vol. 41, no. 314, pp. 256–275, 1950.

[19] M. Campbell, A. J. Hoane, and F. Hsu, "Deep Blue," *Artificial Intelligence*, vol. 134, no. 1–2, pp. 57–83, 2002.

[20] Chacoma and O. V. Billoni, "Emergent Complexity in the Decision-Making Process of Chess Players," *Scientific Reports*, vol. 15, no. 1, 2025, doi: 10.1038/s41598-025-06335-9.

[21] S. Chowdhary, I. Iacopini, and F. Battiston, "Quantifying Human Performance in Chess," *arXiv preprint arXiv:2207.07780*, 2022.

[22] D. H. Minh and T. N. Duc, "Stockfish Chess Engine Software Application to Enhance Opening Playing Skills for Physical Education Students," *International Journal of Human Movement and Sports Sciences*, vol. 13, no. 1, pp. 118–125, 2025, doi: 10.13189/saj.2025.130113.

[23] N. Aemy and M. Al-Husaini, "CHESTEGA: Steganography Using PGN Standard in Web-Based Chess Applications," *Journal of Information Systems and Technology (JustIN)*, vol. 11, no. 3, p. 515, 2023, doi: 10.26418/justin.v11i3.66716.

[24] G. De Marzo and V. D. Servedio, "Quantifying the Complexity and Similarity of Chess Openings Using Online Chess Community Data," *arXiv preprint arXiv:2206.14312*, 2022.

[25] T. N. T. Nainggolan, S. Sudarto, and T. W. Ginting, "Analysis of User Acceptance of the Lichess Application Using the UTAUT3 Model," *Jurnal SIFO Mikroskil*, vol. 25, no. 2, pp. 107–124, 2024, doi: 10.55601/jsm.v25i2.1402.