Tikrit Journal of Administrative and Economic Sciences, Vol. 22, No. 74, Part (1): 817-841
Doi: www.doi.org/10.25130/tjaes.22.74.1.40

Tikrit Journal of Administrative and Economics Sciences
مجلة تكريت للعلوم الإدارية والاقتصادية
EISSN: 3006-9149
PISSN: 1813-1719

Design and Evaluation of a Progressive Web Application for Supporting Group Theory Learning in Higher Education

Mahera R. Qasem*, Murtada Raad Youssef
College of Education for Pure Sciences/ Tikrit University

Keywords:
Group Theory, Abstract Algebra, Mathematics Education, JavaScript, Interactive Learning, Software Evaluation.

A R T I C L E I N F O

Article history:

Received	02 Jan. 2026
Received in revised form	29 Jun. 2026
Accepted	07 May. 2026
Available online	30 Jun. 2026

*Corresponding author:
Mahera R. Qasem
College of Education for Pure Sciences/ Tikrit University

Abstract: This study presents the design, implementation, and evaluation of an interactive Progressive Web Application (PWA) for supporting undergraduate learning and problem-solving in group theory. The application was developed using standard web technologies, including HTML, CSS, JavaScript, service-worker caching, and mathematical rendering libraries. It provides computational and visual modules for modular groups, unit groups, permutation groups, the dihedral group D_4, quotient groups, composition series, solvable groups, Sylow subgroups, infinite groups, educational puzzles, and isomorphism testing. The methodology combines theoretical analysis of group-theory concepts with a systems development procedure that includes requirements analysis, interface design, algorithm construction, testing, and user evaluation. The system was evaluated through a questionnaire completed by 270 participants from different academic levels. The findings indicate high perceived educational effectiveness, interface clarity, reduced manual calculation burden, and strong support for self-directed learning. The proposed system offers a low-cost, installable, offline-capable, and pedagogically oriented tool for improving the practical understanding of abstract algebra concepts. The strongest evaluation result was related to perceived educational effectiveness, which recorded the highest mean score ($M = 4.51$). The most important recommendation is to develop a full Arabic interface, extend the system toward matrix groups, rings, and fields, and examine its learning impact through controlled pre-test/post-test studies.

Tikrit Journal of Administrative and Economic Sciences, Vol. 22, No. 74, Part (1): 817-841
Doi: www.doi.org/10.25130/tjaes.22.74.1.40

تصميم وتقييم تطبيق ويب تقدمي لدعم تعلم نظرية الزمر في التعليم العالي

ماهرة ربيع قاسم **مرتضى رعد يوسف**
كلية التربية للعلوم الصرفة/جامعة تكريت

المستخلص

يهدف هذا البحث إلى تصميم وتنفيذ وتقييم تطبيق ويب تقدّمي تفاعلي لدعم تعلّم طلبة المرحلة الجامعية لموضوع نظرية الزمر وحلّ مسائله الأساسية. اعتمد النظام على تقنيات الويب القياسية، ومنها HTML: وCSS وJavaScript وآلية Service Worker، مع عرض رياضي واضح للرموز والنتائج. يتضمن التطبيق وحدات عملية لمعالجة الزمر المعيارية، وزمر الوحدات، وزمر التبديلات، والزمر الديهدرالية، والزمر خارج القسمة، والسلاسل التركيبية، والزمر القابلة للحل، وزمر سيلو، إضافةً إلى عالم الزمر غير المنتهية، والألعاب التعليمية، واختبار التشاكل. اتبعت الدراسة منهجاً وصفياً تحليلياً ومنهج تطوير النظم، بدءاً من تحليل المشكلة وجمع المتطلبات، ثم تصميم الواجهة والخوارزميات، وانتهاءً بالاختبار والتقييم. أظهرت نتائج الاستبانة، التي شملت 270 مشاركاً، أن النظام أسهم في تسهيل الفهم، وتقليل عبء الحساب اليدوي، ودعم التعلّم الذاتي. وتبرز أهمية النظام في كونه أداة مجانية قابلة للتثبيت والعمل دون اتصال بعد التحميل الأول، ومناسبة لتقريب المفاهيم المجردة في الجبر الحديث. أظهرت نتائج الاستبانة، التي شملت 270 مشاركاً، أن النظام أسهم في تسهيل الفهم، وتقليل عبء الحساب اليدوي، ودعم التعلّم الذاتي؛ إذ سجّل بُعد الفاعلية التعليمية المدركة أعلى متوسط حسابي بلغ 4.51. وتبرز أهمية النظام في كونه أداة مجانية قابلة للتثبيت والعمل دون اتصال بعد التحميل الأول، ومناسبة لتقريب المفاهيم المجردة في الجبر الحديث. وتوصي الدراسة بتطوير واجهة عربية متكاملة، وإضافة وحدات خاصة بزمر المصفوفات والحلقات والحقول، وإجراء دراسات تجريبية لاحقة تقيس أثر النظام في التحصيل الفعلي للطلبة من خلال تصميم قبلي/بعدي مضبوط.

الكلمات المفتاحية: نظرية الزمر، الجبر المجرد، تعليم الرياضيات، جافا سكربت، التعلم التفاعلي، تقييم البرمجيات.

1. Introduction

Abstract algebra is a fundamental area of mathematics because it studies structural properties that appear in number theory, geometry, topology, cryptography, and several branches of applied mathematics. Within abstract algebra, group theory provides a rigorous language for symmetry, operations, algebraic invariants, and transformations (Dummit & Foote, 2004; Gallian, 2021; Kleiner, 2007; Rotman, 1995). However, the same abstraction that gives group theory its theoretical power often creates a pedagogical obstacle for undergraduate students. Students are required to reason about closure, associativity, identities, inverses, subgroups, quotient structures, generators, permutations, and composition series. At the same

time, many of these objects are not directly visible in the traditional classroom setting.

A repeated difficulty in teaching group theory is the gap between formal definitions and practical computation. Manual construction of Cayley tables, element orders, cosets, quotient groups, or permutation compositions is time-consuming and error-prone, especially when groups of larger order are involved. Advanced systems such as GAP and SageMath provide powerful algebraic computation, but they are primarily research-oriented and usually require command-line use or programming experience (The GAP Group, 2025; The SageMath Developers, 2026). For undergraduate students, this can create a second abstraction barrier: the learner must understand both the mathematical structure and the syntax of the software.

The present study addresses this gap by designing a browser-based educational application that performs essential group-theory computations while presenting results in a clear visual and mathematical form. The system follows an educationally oriented design in which computational output, symbolic representation, and immediate feedback are combined to translate abstract structures into inspectable representations (Anderson, 2008; Tall, 2004).

The motivation for this study arises from three interconnected needs. First, undergraduate students often encounter group theory as a highly symbolic subject in which the relation between formal definitions and concrete examples is not immediately visible. Second, manual computation of element orders, cosets, quotient structures, permutation products, and Cayley tables can become lengthy and error-prone, which may shift students' attention from conceptual reasoning to mechanical calculation. Third, although advanced computational algebra systems provide extensive algebraic capabilities, their command-based nature may not be suitable for beginners who need guided interaction, immediate feedback, and visually organized mathematical output.

Accordingly, the objective of this research is to design, implement, and evaluate an interactive Progressive Web Application that supports undergraduate learning of group theory by combining accurate computation, visual representation, LaTeX-quality mathematical output, and offline accessibility. The application is not intended to replace comprehensive research-level algebra systems; rather, it is designed as a pedagogical tool

that helps students inspect algebraic structures, verify calculations, and connect theoretical definitions with practical examples.

The scientific contribution of the study lies in presenting a browser-based educational model for abstract algebra learning that integrates group-theoretic algorithms, progressive web application architecture, interactive visualization, and user-based evaluation. Unlike general-purpose algebra systems, the proposed system focuses on learning support, reduced entry barriers, and structured feedback for undergraduate students. This contribution is practical and pedagogical rather than a claim of superiority over established computational algebra systems.

The remainder of this paper is organized as follows. Section 2 reviews the relevant literature on group-theory learning, computational algebra systems, digital mathematics learning, and progressive web applications. Section 3 presents the theoretical framework and the algebraic structures implemented in the system. Section 4 describes the methodology, requirements, architecture, and validation procedures. Section 5 presents the practical interfaces and evaluation results. Section 6 concludes the study and outlines future development directions.

2. Literature Review

2-1. Group theory learning and the abstraction problem: Group theory occupies a central position in abstract algebra because it introduces students to algebraic structures, symmetry, transformations, and invariant properties. Standard references emphasize the formal construction of groups, subgroups, normal subgroups, quotient groups, permutation groups, and homomorphisms (Dummit & Foote, 2004; Gallian, 2021; Rotman, 1995). However, the educational difficulty of group theory does not arise only from the definitions themselves, but also from the need to coordinate symbolic reasoning with concrete computation. Students must often move between formal statements and examples such as modular groups, unit groups, symmetric groups, and dihedral groups. This transition is pedagogically demanding because abstract algebra requires learners to operate with objects that are not directly visual.

2-2. Computational algebra systems and educational accessibility: Computational algebra systems such as GAP and SageMath provide extensive support for advanced algebraic computation. GAP is particularly strong in computational discrete algebra and computational group theory,

while SageMath integrates several open-source mathematical packages into a broad computational environment. These systems are valuable for researchers and advanced users; however, beginner students may face difficulties related to syntax, command structure, and the need to understand both the mathematics and the software environment. Therefore, there remains a need for educational interfaces that present selected group-theory computations in a guided, visual, and low-barrier format.

2-3. Digital and visual learning in mathematics: Research in mathematics education indicates that visual representation, interaction, and feedback can support conceptual development when they are aligned with the mathematical structure being taught (Anderson, 2008; Mayer, 2020; Tall, 2004). In the context of abstract algebra, visual and computational representations can help students inspect the results of operations, observe generated subgroups, compare cosets, and verify structural properties. The educational value of technology therefore depends not merely on automation, but on how clearly it connects computational output with mathematical meaning.

2-4. Progressive Web Applications as an educational architecture: Progressive Web Applications provide a suitable technical framework for educational software because they can be accessed through a browser, installed on different devices, and used offline after caching essential files through a service worker. This architecture is particularly useful in educational settings where students may use different devices or have unstable internet access. JavaScript, together with mathematical rendering libraries such as MathJax or KaTeX, allows interactive computation and professional display of mathematical expressions directly in the browser. For this reason, a PWA is an appropriate architecture for a lightweight, accessible, and device-independent group-theory learning tool.

2-5. Evaluation of educational software: The evaluation of educational software should distinguish between perceived usefulness, usability, clarity, and actual learning impact. Questionnaire-based evaluation can provide descriptive evidence about users' perceptions, but it cannot by itself prove causal improvement in achievement. Therefore, the present study uses questionnaire results to evaluate perceived educational usefulness, interface clarity, and reduction of manual computational burden, while

acknowledging that future controlled studies are needed to measure learning gains directly.

The literature on mathematical software and digital learning indicates that computational environments can support conceptual understanding when they are designed for educational interaction rather than merely for advanced technical computation. In abstract algebra, this distinction is particularly important because students must connect formal definitions with concrete examples and verifiable calculations.

Existing algebra systems such as GAP and SageMath are mathematically powerful and suitable for advanced computation (The GAP Group, 2024; The Sage Developers, 2024). Their strength is the breadth and depth of symbolic algebra, but their educational limitations are clear for beginner users: they generally require syntax knowledge, programming habits, and a degree of abstraction that may exceed the immediate needs of students.

Progressive Web Applications provide an appropriate architecture for this purpose. A PWA can be opened through a browser, installed on desktop or mobile devices, and used offline after caching core files through a service worker (MDN Web Docs, 2025). JavaScript is also suitable for lightweight mathematical applications because computation, input validation, and dynamic display can all be performed in the client browser (Flanagan, 2020).

3. **The theoretical framework**: A group is an algebraic structure $(G,*)$ satisfying closure, associativity, identity, and inverse axioms. The application focuses on structures frequently encountered in an undergraduate abstract algebra course. The modular additive group is defined by $Z_n = \{0, 1, 2, \dots, n-1\}$, with operation $a +_n b = (a + b) \bmod n$.

For $a \in Z_n$, the order of the element is validated by the standard formula $\mathrm{ord}_{Z_n}(a) = n/\gcd(a, n)$. The unit group modulo n is represented as $U(n) = \{a \in Z_n : \gcd(a, n) = 1\}$, with multiplication modulo n.

The system also supports the symmetric group S_n, where elements are permutations composed as functions, and the dihedral group D_4, the group of eight symmetries of a square. For quotient structures, if N is a normal subgroup of G, the coset set G/N forms a group under $(gN)(hN) = (gh)N$. These topics combine conceptual abstraction with procedural calculation and therefore provide an appropriate basis for interactive computational feedback.

The theoretical basis of the application therefore combines two dimensions. The first is algebraic correctness, where every output must conform to standard definitions of groups, subgroups, cosets, quotient groups, generators, element orders, and isomorphisms. The second is pedagogical representation, where the same output must be displayed in a form that students can inspect, compare, and verify visually.

4. **Methodology and Data**: This project employs a descriptive-analytical approach to studying group theory concepts and a systems development approach to designing and building an interactive platform for solving group theory problems. This design-oriented structure is consistent with information-systems research in which an artefact is developed, tested, and evaluated against a practical problem (Hevner et al., 2004). This procedure is also consistent with design science research methodology, in which a relevant problem is identified, an artefact is designed and developed, and its usefulness is evaluated in relation to the intended context of use (Peffers et al., 2007). The platform aims to simplify the solution of computational problems related to groups, reduce the difficulty of dealing with abstract concepts, and help students quickly and accurately verify their solutions.

This procedure is also consistent with design science research methodology, in which a relevant problem is identified, an artefact is designed and developed, and its usefulness is evaluated in relation to the intended context of use (Peffers et al., 2007). Accordingly, the present study treats the proposed PWA as an educational artefact whose value is assessed through technical validation and user-based descriptive evaluation.

4-1. Work Stages

4-1-1. Problem Analysis: Many students find it difficult to solve abstract algebra problems, especially when dealing with large-order groups, due to the abstract nature of the material and the difficulty of applying it practically. Traditional teaching methods rely heavily on manual computation, which is time-consuming and prone to error. Furthermore, existing computational tools such as GAP and SageMath, while powerful, require programming experience and are not designed with the undergraduate student in mind.

Therefore, this platform was designed to be a tool that helps students solve problems quickly and without computational errors. It also allows researchers to test complex examples without the need for lengthy manual calculations, thus facilitating understanding of the material and improving

self-study. The decision to implement the system as a Progressive Web Application (PWA) was motivated by the need to reach the widest possible audience, including students on mobile devices with limited or no internet connectivity.

4-1-2. Requirements Gathering

4-1-2-1. Functional Requirements:

A. Group Problem Solving: The system computes the order of a group, the order of each element, subgroups, generators, modular arithmetic, composition of permutations, quotient groups, cosets, composition series, solvable groups, Sylow subgroups, and isomorphism testing.

B. LaTeX-Quality Mathematical Rendering: All mathematical output is displayed using LaTeX-quality typography, ensuring professional and unambiguous presentation of results to the user (MathJax Consortium, 2025).

C. Animated Theorem Visualisation: The Jordan-Hölder theorem module features animated composition series with colour-coded highlighting that identifies the subgroup violating a series condition.

D. Infinite Groups World: A dedicated interactive environment guides students through infinite algebraic structures including $(Z,+)$, $(Q,+)$, $(R,+)$, matrix groups, and $(C,+)$, each with its own interactive visualisation.

E. Educational Games and Puzzles: An integrated games section within the Infinite Groups World provides interactive challenges related to group-theoretic concepts.

F. Isomorphism Tester: The system allows the user to select two groups and test whether they are isomorphic, displaying a justification for the result.

G. Data Input and Results Display: The user can easily enter group parameters, and results are presented immediately in a well-organised format.

4-1-2-2. Non-Functional Requirements:

A. Ease of Use: The platform features a clear and simple interface suitable for users with no programming experience.

B. Accuracy: All calculations are verified to ensure precise results consistent with theoretical definitions.

C. Speed: The platform solves problems instantly without unnecessary delays.

D. Offline Capability: The PWA architecture allows the platform to function fully after the first load, with no internet connection required.

E. Installability: The platform can be installed on any smartphone, tablet, or desktop directly from the browser, free of charge and without requiring an app store.
F. Scalability: The system can be extended in the future to include additional algebraic theories such as ring theory and field theory.

4-1-2-3. Technical Requirements

- HTML and CSS: Used to build and style the user interface.
- JavaScript: Used to implement all algorithms and perform group-related operations.
- Service Worker: Implements the PWA offline caching strategy.
- Web App Manifest: Enables installation on mobile and desktop devices.
- MathJax / KaTeX: Provides LaTeX-quality mathematical rendering within the browser.
- Device Compatibility: The platform runs on desktop computers, tablets, and mobile devices across all major operating systems.

4-2. System Design

4-2-1. Overall Architecture: The platform is designed as a client-side single-page application with no server-side dependencies. All computations are performed locally within the user's browser, which guarantees privacy, speed, and offline functionality.

The system first receives the group parameters or the operation that the user wants to perform. It then validates the input to ensure correctness and freedom from errors. After validation, it performs the requested calculations and displays the results immediately in a structured format. This workflow ensures accurate and fast performance while maintaining a simple interface.

The PWA architecture adds two key components on top of the standard web application:

A Service Worker that intercepts network requests and serves cached assets when the device is offline.

A Web App Manifest that declares the application name, icons, theme colours, and display mode, enabling the browser to offer installation to the user.

4-2-2. Flowchart: The following flowchart summarises the main steps of the system from input validation to result display.

Doi: www.doi.org/10.25130/tjaes.22.74.1.40

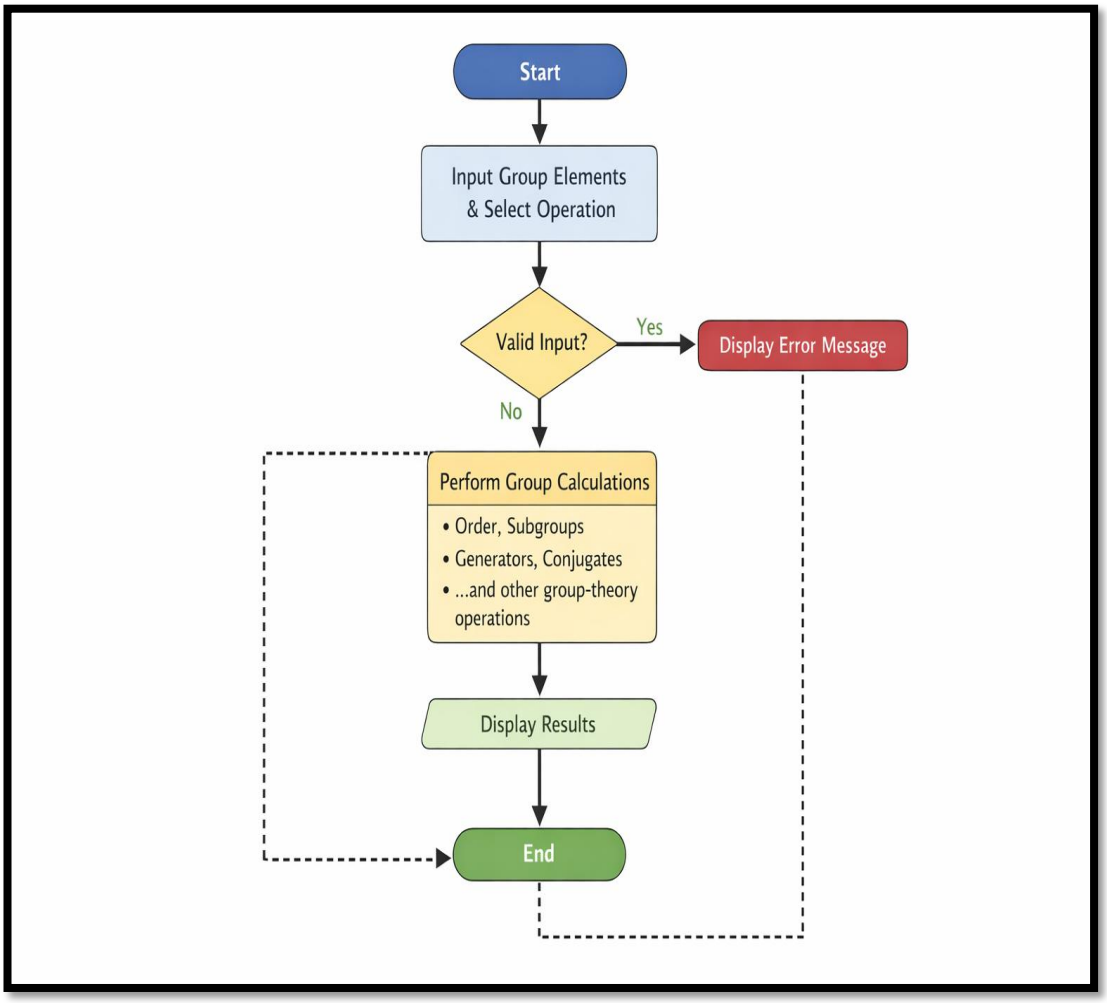

Figure (1): System flowchart from user input to results display

Source: Prepared by the researcher based on the system design.

4-2-3. Algorithm Design: Each group operation is implemented using JavaScript functions that follow the definitions and rules of group theory. The platform includes algorithms to compute:

- ❖ The order of a group and the order of each element.
- ❖ Subgroups and their generators.
- ❖ Cosets, normal subgroups, and quotient groups.
- ❖ Composition series and solvable groups.
- ❖ Sylow p-subgroups.
- ❖ Isomorphism between groups.

All algorithms are structured to minimise errors and keep calculations fast and reliable. Inputs are processed step by step, and results are computed according to well-defined rules derived from mathematical theory. As an

illustration, Algorithm 1 shows the procedure for computing the order of an element in Z_n.

Algorithm 1. Computing the additive order of an element in $\mathbb{Z}_n$

Input: $a \in \mathbb{Z}_n$, modulus n

Output: $\text{ord}_{\mathbb{Z}_n}(a)$

Step	Procedure
1	$a \leftarrow a \bmod n$
2	If $a = 0$, return 1
3	$k \leftarrow 1$
4	$x \leftarrow a$
5	While $x \neq 0$, do:
6	$x \leftarrow (x + a) \bmod n$
7	$k \leftarrow k + 1$
8	End while
9	Return k

Explanatory note: This algorithm is consistent with the additive structure of $\mathbb{Z}_n$, where the identity element is 0. For unit groups U(n), the multiplicative order is computed separately by repeated modular multiplication until the identity element 1 is reached.

4-2-4. User Interface Design: The interface is designed to be simple and intuitive. A collapsible sidebar navigation menu on the left provides access to all major sections of the application. When a menu item is selected, the corresponding tools and visualisations appear in the main display area on the right. Key design principles applied throughout the interface are as follows:

- ❖ Mathematical clarity: All group elements, operations, and results are rendered with LaTeX-quality typography.
- ❖ Visual feedback: Colour coding and animation are used to highlight important structural features, such as the subgroup that breaks a composition series.
- ❖ Progressive disclosure: Basic results are shown immediately; additional details such as full element lists and Cayley tables are available on demand via expandable panels.
- ❖ Responsive layout: The interface adapts to all screen sizes, from large desktop monitors to small smartphone screens.

Doi: www.doi.org/10.25130/tjaes.22.74.1.40

4-2-5. Architecture Diagram: The architecture diagram summarises the interaction between the user interface, input validation, processing layer, and output layer.

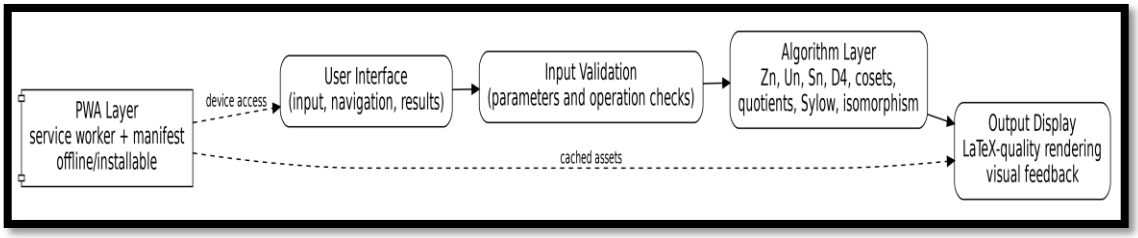

Figure (2): System architecture diagram showing the four main components.

Source: Prepared by the researcher based on the developed application.

The system consists of four main components:

A. User Interface (UI): Allows users to input group parameters and select operations.

B. Input Validation Layer: Checks that all inputs are correct before any computation is performed.

C. Processing Unit (Algorithm Layer): Executes all group-theory calculations according to the defined algorithms.

D. Output Display Layer: Renders results immediately in a structured and readable format with LaTeX-quality mathematics.

4-2-6. Comparison with Other Systems: Table 1 compares the proposed system with the well-known mathematical software systems GAP and SageMath. While GAP and SageMath are powerful computational algebra systems designed for advanced mathematical research, the proposed system focuses primarily on educational support and ease of use for students studying group theory.

Table (1): Pedagogical and technical comparison between the proposed educational PWA, GAP, and SageMath

Feature / Criterion	Proposed System	GAP	SageMath
Main purpose	Undergraduate learning support	Advanced computational algebra	Broad mathematical computation
Research-level algebra depth	Limited	High	High

Doi: www.doi.org/10.25130/tjaes.22.74.1.40

Feature / Criterion	Proposed System	GAP	SageMath
Need for programming or commands	No for supported modules	Yes	Moderate to high
Guided educational interface	High	Limited	Moderate through notebooks
Browser-based use without CAS installation	Yes	No	Partial
Offline use	Yes, after first load	Yes, after local installation	Yes, after local installation
Mobile install ability as PWA	Yes	No native PWA	No native PWA
Group-theory computational power	Moderate	High	High
Visual/animated theorem support	Yes	Not a core feature	Not a core feature
Suitability for beginners	High	Moderate to low	Moderate

Source: Prepared by the researcher based on the pedagogical and technical comparison of the proposed system with existing software.

The comparison should be interpreted in terms of educational purpose rather than computational superiority. GAP and SageMath offer deeper, broader algebraic capabilities and are better suited to advanced research-level computation. In contrast, the proposed system is intentionally limited to selected undergraduate group theory topics. It is designed to reduce the entry barrier for students through guided interaction, visual feedback, mobile install ability, and offline browser-based use. Therefore, the proposed PWA complements rather than replaces established computational algebra systems.

4-3. Testing and Validation

4-3-1. Testing Steps

A. Input Validation: The system checks that all group parameters are correct and that the requested operations are suitable for the selected group type.

B. 2. Calculation Verification: The system was tested using standard and well-known examples in group theory. Operations on Z_n, S_3, D_4, and several quotient groups were verified to ensure correct results for group order, subgroups, generators, cosets, composition series, and solvability.

C. LaTeX Rendering Verification: All mathematical output was inspected to confirm that symbols, fractions, subscripts, and superscripts are rendered correctly across different browsers and screen sizes.

D. PWA and Offline Testing: The application was installed on Android and iOS devices and tested in airplane mode to verify that all modules function correctly without an internet connection.

E. Performance Testing: Operations were executed on higher-order groups to verify that the platform produces results promptly. Computational speed may vary depending on the user's device, as more complex groups require more processing power.

F. User Interface Testing: The interface was evaluated across multiple devices and screen sizes to ensure clarity, responsiveness, and ease of use.

4-3-2 Testing Goals:

- ❖ Ensure that all calculations are accurate and consistent with theoretical results.
- ❖ Confirm that the platform is free of calculation and programming errors.
- ❖ Verify that the PWA installation and offline functionality work correctly on both Android and iOS devices.
- ❖ Confirm that LaTeX rendering is consistent and correct across all supported browsers.
- ❖ Verify that the platform is fast, reliable, and easy to use across different devices.

5. Practical Application and Evaluation Results

5-1. Practical application interfaces: This section presents the practical interfaces extracted from the developed application. The figures are arranged as multi-panel images to preserve the complete set of screenshots while keeping the article concise and suitable for journal submission.

Doi: www.doi.org/10.25130/tjaes.22.74.1.40

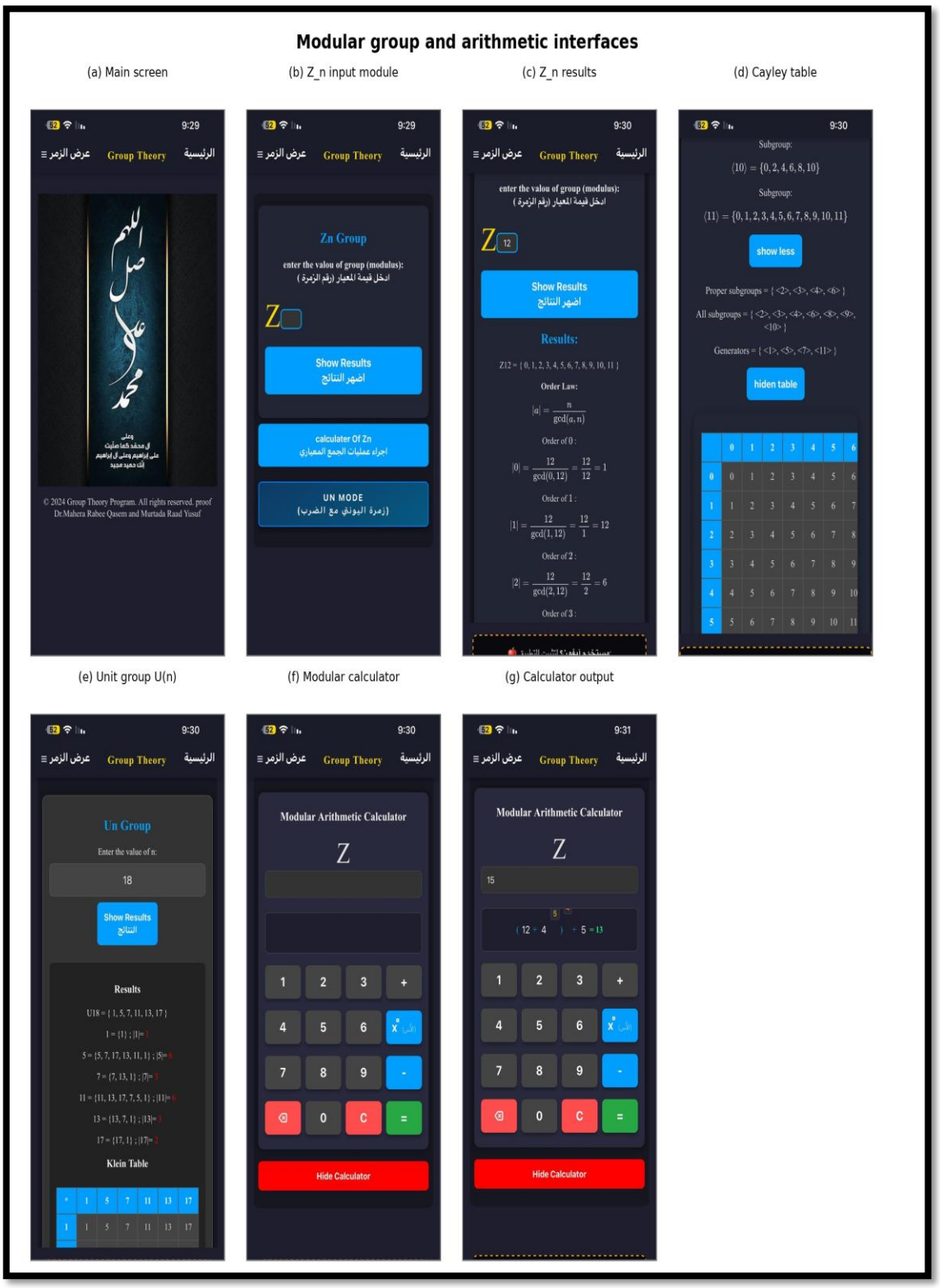

Figure (3): Modular group, unit group, Cayley table, and modular arithmetic interfaces

Source: Prepared by the researcher based on screenshots from the developed application.

Doi: www.doi.org/10.25130/tjaes.22.74.1.40

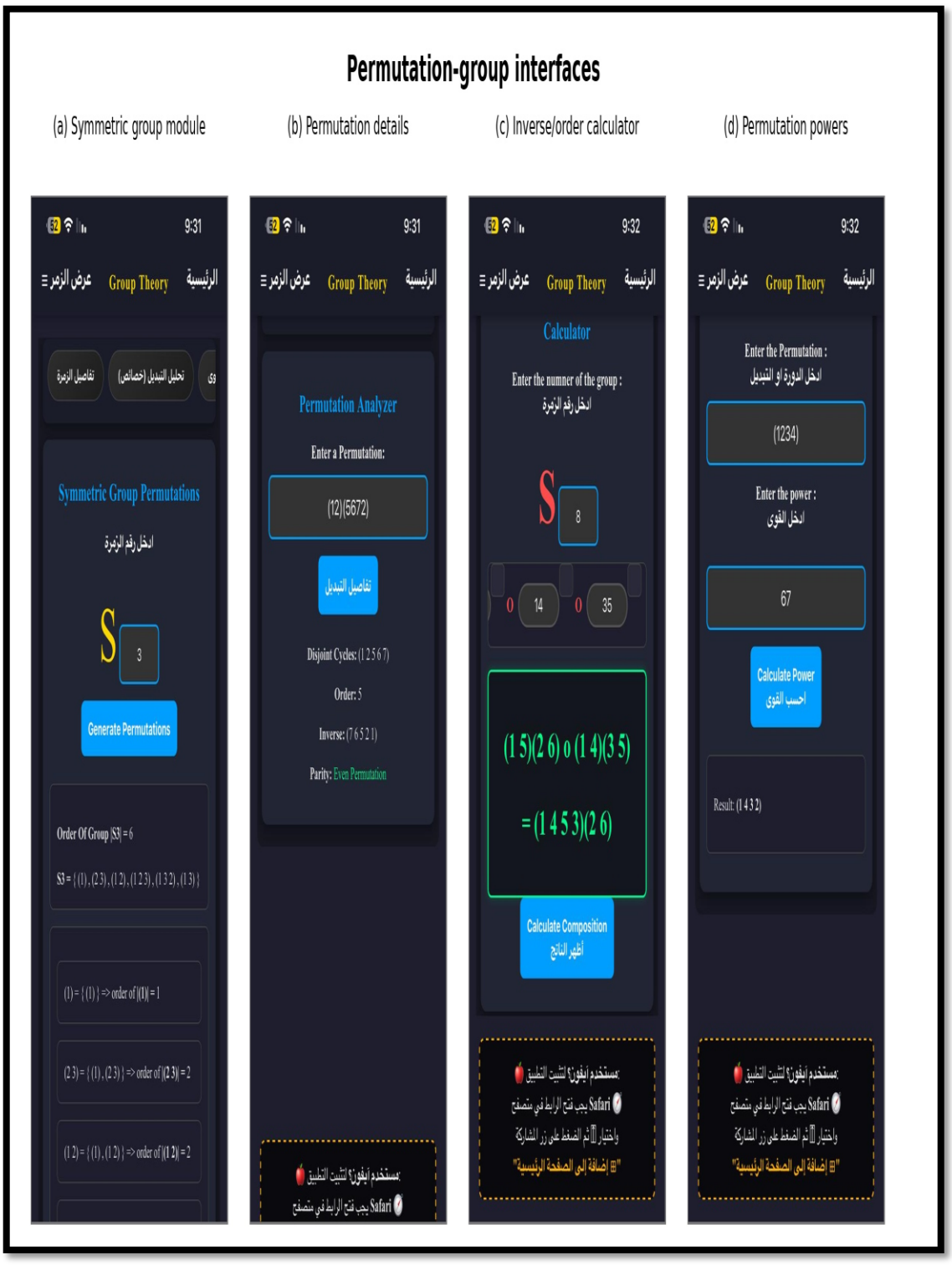

Figure (4): Symmetric group interfaces including permutation details, inverse/order calculation, and powers.

Source: Prepared by the researcher based on screenshots from the developed application.

Doi: www.doi.org/10.25130/tjaes.22.74.1.40

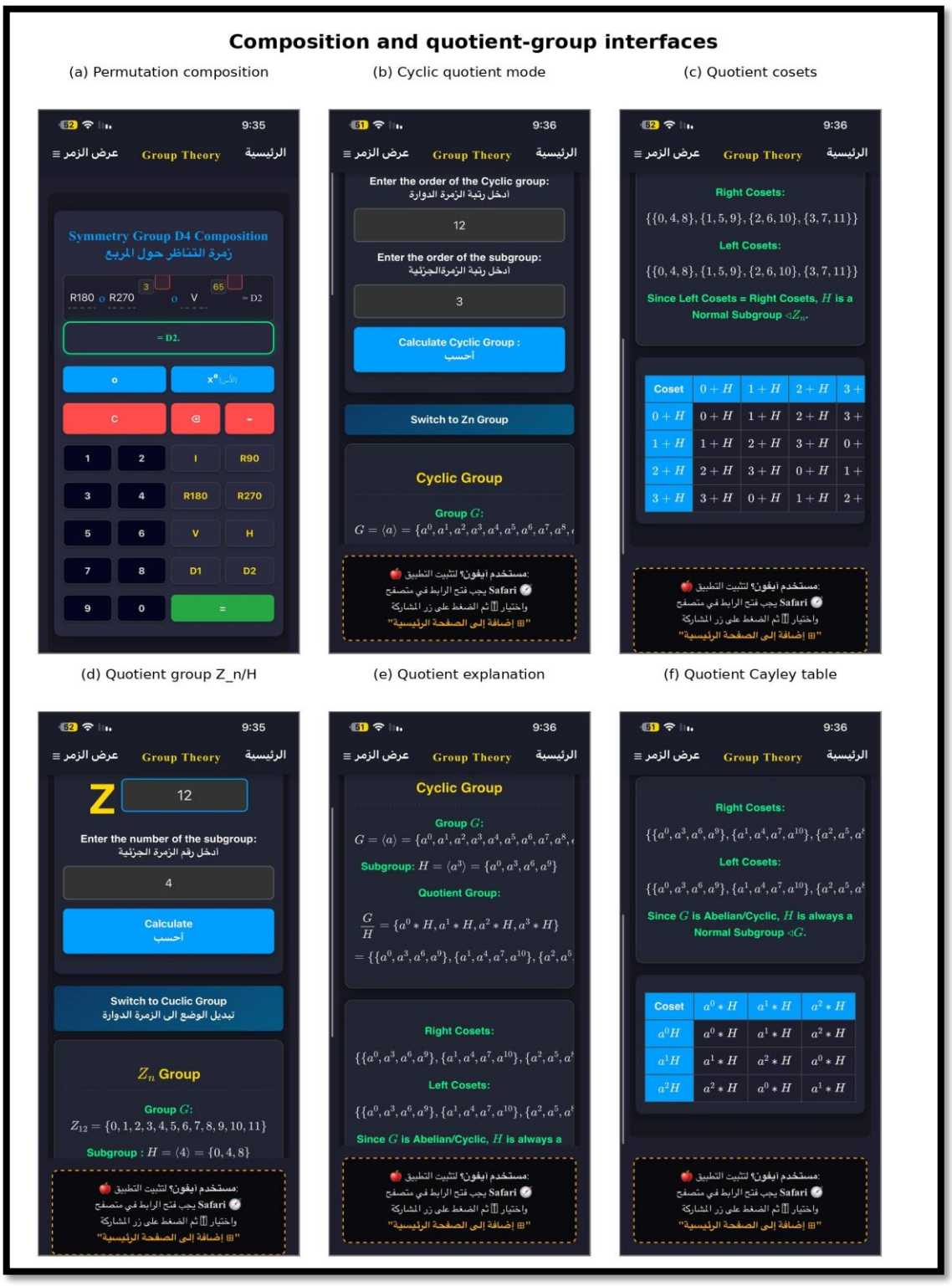

Figure (5): Permutation composition and quotient-group construction interfaces

Source: Prepared by the researcher based on screenshots from the developed application.

Doi: www.doi.org/10.25130/tjaes.22.74.1.40

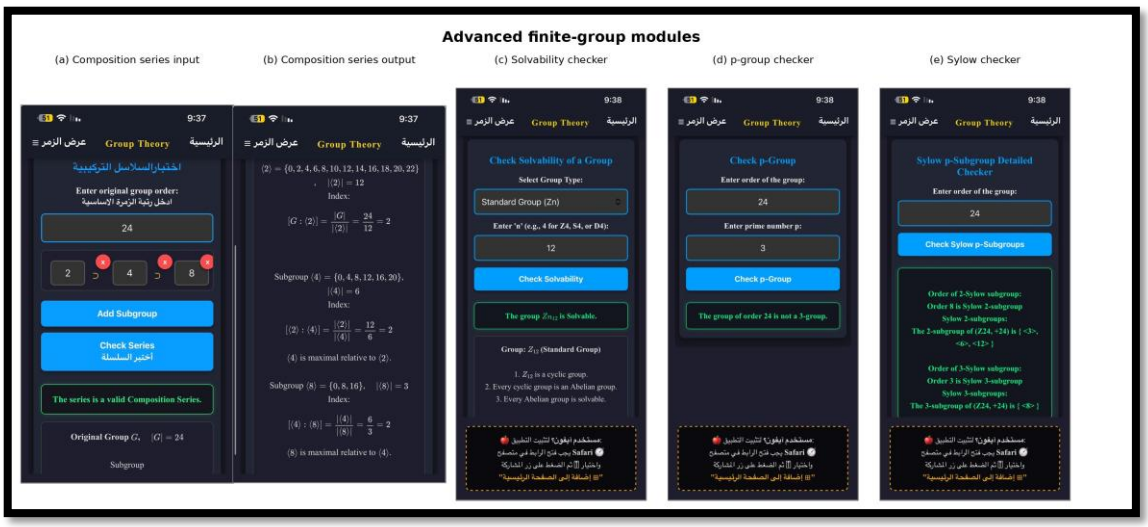

Figure (6): Advanced finite-group modules: composition series, solvability, p-groups, and Sylow subgroups

Source: Prepared by the researcher based on screenshots from the developed application.

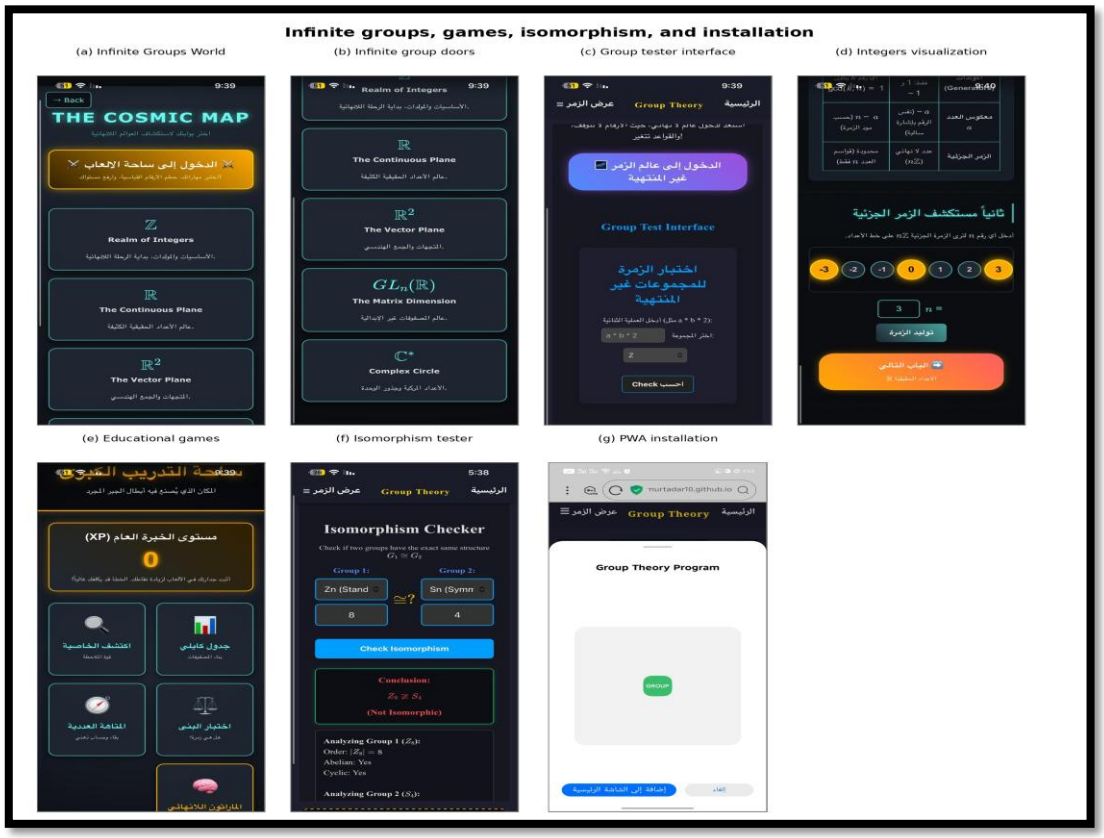

Figure (7): Infinite Groups World, educational games, isomorphism tester, and PWA installation interface

Source: Prepared by the researcher based on screenshots from the developed application.

5-2. Questionnaire design and participant profile: The questionnaire instrument is presented in Appendix A to improve methodological transparency. In future extensions of the study, the internal consistency of the Likert-scale items should also be examined, with careful interpretation of Cronbach's alpha as a reliability indicator rather than as conclusive proof of instrument validity (Taber, 2018). The analysis is descriptive and reports frequencies, percentages, means, and agreement levels. Since the study did not include a control group or pre-test/post-test design, the questionnaire results are interpreted as evidence of perceived usefulness and usability rather than causal evidence of learning improvement.

The application was evaluated using a structured electronic questionnaire after participants had interacted with the system. The questionnaire targeted undergraduate students from different academic levels, postgraduate students, and specialist faculty members in mathematics. A total of 270 participants completed the questionnaire. The instrument contained demographic questions, a manual problem-solving difficulty item, eight Likert-scale statements, and open-ended questions about strengths, technical difficulties, and suggested improvements.

The difficulty distribution shows that 165 participants, or 61.1%, rated manual problem solving as difficult or very difficult. This supports the research premise that group-theory computation creates a real instructional burden when students rely only on manual methods.

Table (2): Participant profile and manual problem-solving difficulty

Measure	Category	Count	Percentage
Academic level	Second year	68	25.2%
	Third year	61	22.6%
	Fourth year	98	36.3%
	Postgraduate / graduate	43	15.9%
Prior experience	Currently studying group theory	72	26.7%
	Previously completed group theory	148	54.8%
	Not yet studied group theory	50	18.5%
Manual difficulty	Very easy	18	6.7%

Doi: www.doi.org/10.25130/tjaes.22.74.1.40

Measure	Category	Count	Percentage
	Easy	29	10.7%
	Moderate	58	21.5%
	Difficult	87	32.2%
	Very difficult	78	28.9%

Source: Prepared by the researcher based on questionnaire responses.

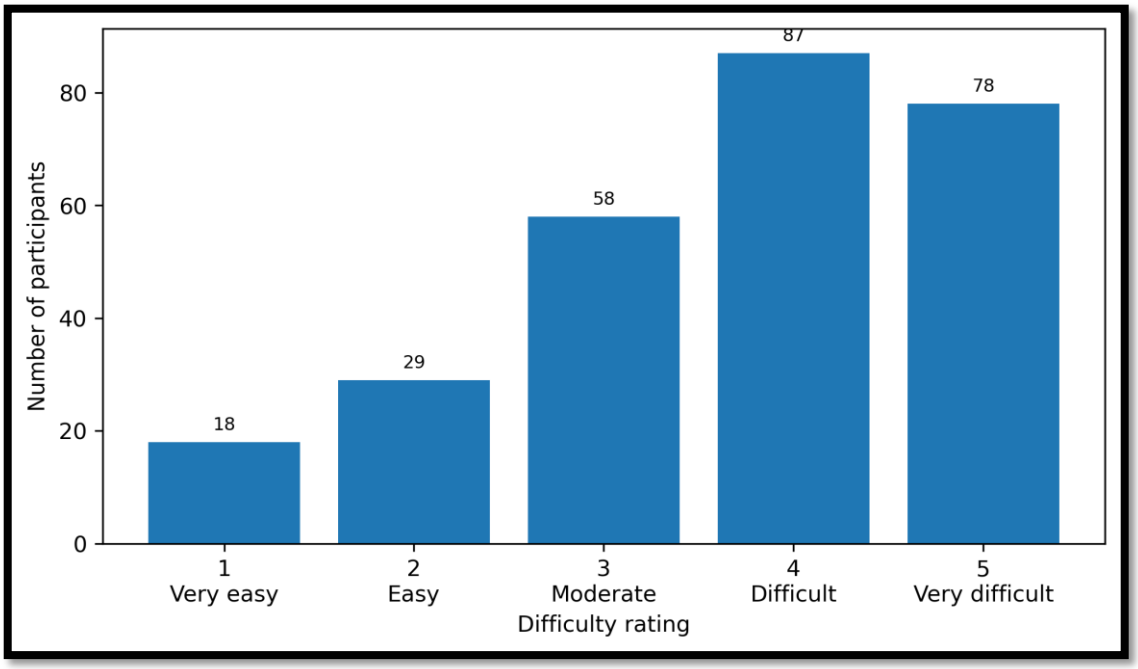

Figure (8): Distribution of self-reported difficulty ratings for manual group-theory problem solving

Source: Prepared by the researcher based on questionnaire responses.

5-3. Likert-scale results: Each statement was scored on a five-point scale from 1 (strongly disagree) to 5 (strongly agree). Table 3 reports the mean score and the percentage of participants who selected either agree or strongly agree. Because Likert-type responses represent ordered response categories, the results were interpreted descriptively using means, percentages, and agreement levels, while avoiding unsupported causal claims about learning improvement (Sullivan & Artino, 2013). Because Likert-type responses represent ordered response categories, the results were interpreted descriptively using means, percentages, and agreement levels, while avoiding unsupported causal claims about learning improvement (Sullivan & Artino, 2013).

Table (3): Summary of Likert-scale evaluation results

No.	Evaluation statement	Mean	Agree / strongly agree	Assessment
S1	The program is a helpful and effective educational tool.	4.51	89.6%	Excellent
S2	Visual and interactive representation deepened my understanding.	4.38	86.3%	Excellent
S3	The program helped me connect theory with practical applications.	4.29	84.1%	Very good
S4	Data entry and results display are organized and fast.	4.33	85.2%	Very good
S5	The user interface is simple and clear.	4.47	88.5%	Excellent
S6	I make errors when solving large groups by hand.	4.22	82.6%	Very good
S7	Manual calculations take a long time and considerable effort.	4.41	87.0%	Excellent
S8	Abstract concepts are difficult without interactive tools.	4.18	81.5%	Very good
Overall	Overall mean and agreement	4.35	85.6%	Excellent

Source: Prepared by the researcher based on questionnaire responses.

Doi: www.doi.org/10.25130/tjaes.22.74.1.40

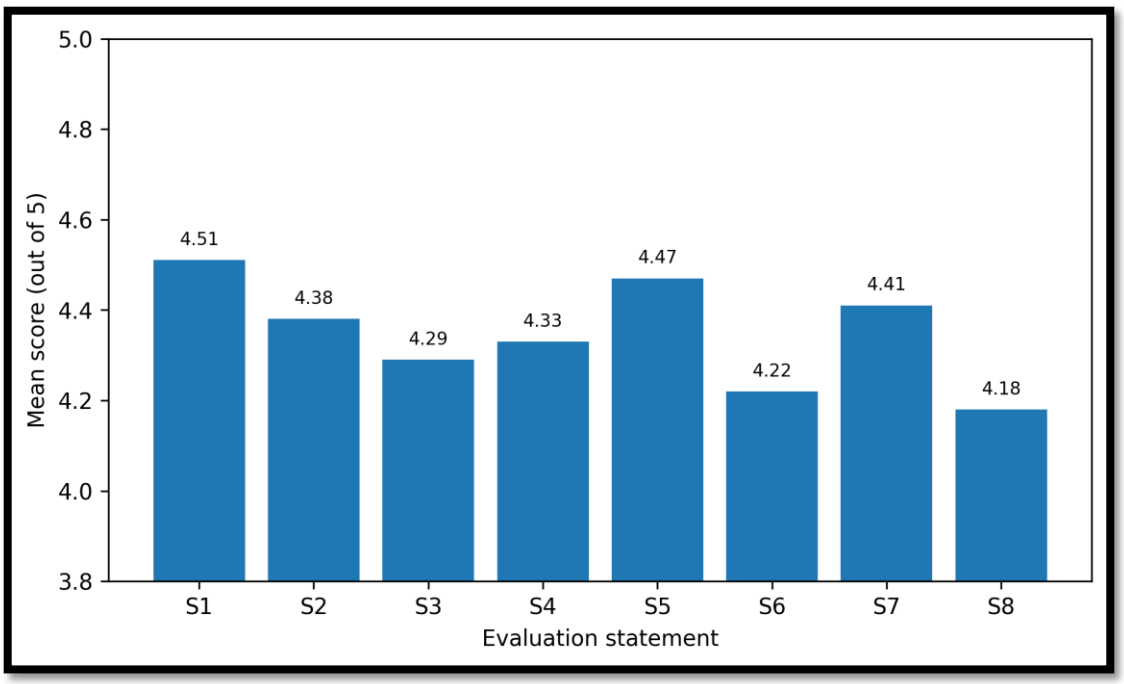

Figure (9): Mean scores for the eight Likert-scale evaluation statements
Source: Prepared by the researcher based on questionnaire responses.

The highest mean score was recorded for educational effectiveness (S1 = 4.51), followed by interface clarity (S5 = 4.47) and the burden of manual calculation (S7 = 4.41). These results indicate that participants valued both the usability and the instructional function of the application. The overall mean of 4.35 suggests a strong positive evaluation, but the results should be interpreted as descriptive evidence rather than proof of causal learning improvement, because the study did not include a controlled pre-test/post-test design.

5-4. Open-ended responses: The most frequently appreciated features were the modular arithmetic calculator, Cayley table generator, fast calculation, readable mathematical output, Infinite Groups World, animated composition-series visualization, and offline mobile installation. Suggested improvements included adding matrix group modules, extending the platform to ring theory and field theory, providing an Arabic-language interface, and expanding theorem visualizations beyond Jordan-Hölder theory. Technical difficulties were limited; a small number of participants needed time to understand the permutation input format, and one participant reported slow performance on an older device when processing larger permutation groups.

5-5. Discussion: The evaluation results confirm that the proposed PWA addresses a practical educational need. Manual computation in group theory is not only lengthy but also structurally complex. When students construct Cayley tables, determine element orders, or test coset operations by hand, a minor arithmetic error can obscure the underlying algebraic concept. The proposed system reduces this procedural burden by making the computation immediate and by presenting the output in a readable mathematical form.

The results also support the value of visual and interactive representation. Statement S2 achieved a mean of 4.38, indicating that participants perceived visualization as helpful for understanding algebraic structure. This is important because group theory often becomes difficult when students cannot connect symbolic definitions with examples. The application creates this connection by showing the effect of operations, generated subgroups, quotient structures, and composition-series conditions directly.

From a software-design perspective, the PWA architecture is one of the main strengths of the project. It avoids installation barriers, app-store restrictions, and server dependence. This is particularly useful in local educational settings where internet connectivity and device resources may differ between students. The approach also demonstrates that an academically useful mathematical tool can be built with standard web technologies and distributed at no cost.

However, the study has limitations. The evaluation is based on self-reported perceptions rather than direct measurement of learning gains. The sample is broad and useful for descriptive evaluation, but a future study should include a controlled experimental design comparing pre-test and post-test performance between students using the system and students relying only on traditional methods. In addition, algorithmic performance for very large groups should be optimized before expanding the system toward research-level computation.

6. Conclusions: This study designed, implemented, and evaluated an interactive Progressive Web Application for supporting undergraduate learning and computation in group theory. The system covers selected undergraduate topics, including modular groups, unit groups, symmetric groups, dihedral groups, quotient groups, composition series, solvable groups, Sylow subgroups, infinite-group examples, educational puzzles, and

isomorphism testing. Its main practical features are immediate computational feedback, LaTeX-quality mathematical rendering, visual interaction, mobile install ability, and offline use through PWA architecture. The questionnaire results from 270 participants indicate positive perceived usefulness, interface clarity, and reduced manual computational burden. The overall mean score was 4.35 out of 5, with an agreement level of 85.6%, while perceived educational effectiveness recorded the highest mean score (M = 4.51). These results suggest that the application is promising as a pedagogical support tool for group-theory learning. However, they should be interpreted as descriptive evidence based on user perceptions rather than definitive proof of causal learning improvement.

The study contributes a practical educational model for transforming selected abstract algebra topics into interactive, browser-based learning modules. It also demonstrates how standard web technologies can be used to build a low-cost and accessible mathematical learning tool. Nevertheless, the current system remains limited in scope and does not compete with research-level computational algebra systems such as GAP and SageMath.

Future work should focus on developing a full Arabic interface, adding matrix groups, rings, and fields, optimizing algorithms for larger groups, and enabling export of results to PDF or LaTeX. More importantly, future research should conduct controlled pre-test/post-test studies to measure actual learning gains and compare the performance of students who use the system with those who rely only on traditional instruction.

Appendix A. Questionnaire Instrument:

A. Academic level: second year / third year / fourth year / postgraduate or graduate.
B. Prior experience with group theory: currently studying / previously completed / not yet studied.
C. Difficulty of solving group-theory problems manually: very easy/easy/moderate/difficult / very difficult.
D. Likert-scale statements, from 1 = strongly disagree to 5 = strongly agree:

- S1. The program is a helpful and effective educational tool.
- S2. Visual and interactive representation deepened my understanding.
- S3. The program helped me connect theory with practical applications.
- S4. Data entry and results display are organized and fast.
- S5. The user interface is simple and clear.

- ❖ S6. I make errors when solving large groups by hand.
- ❖ S7. Manual calculations take a long time and considerable effort.
- ❖ S8. Abstract concepts are difficult without interactive tools.
- ❖ Open-ended questions: What are the strongest features of the application? What technical difficulties did you face? What improvements do you suggest?

Reference

1. Anderson, T. (Ed.). (2008). The theory and practice of online learning (2nd ed.). Athabasca University Press.
2. Dummit, D. S., & Foote, R. M. (2004). Abstract algebra (3rd ed.). John Wiley & Sons.
3. Flanagan, D. (2020). JavaScript: The definitive guide (7th ed.). O'Reilly Media.
4. Gallian, J. A. (2021). Contemporary abstract algebra (10th ed.). CRC Press. doi: 10.1201/9781003142331
5. Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. MIS Quarterly, 28(1), 75–105. doi: 10.2307/25148625
6. Kleiner, I. (2007). A history of abstract algebra. Birkhäuser. doi: 10.1007/978-0-8176-4685-1
7. MathJax Consortium. (2025). MathJax documentation. Retrieved June 28, 2026, from https://docs.mathjax.org
8. Mayer, R. E. (2020). Multimedia learning (3rd ed.). Cambridge University Press.
9. MDN Web Docs. (2025, August 25). Progressive web apps. Mozilla. Retrieved June 28, 2026, from https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps
10. Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. Journal of Management Information Systems, 24(3), 45–77. doi: 10.2753/MIS0742-1222240302
11. Rotman, J. J. (1995). An introduction to the theory of groups (4th ed.). Springer. doi: 10.1007/978-1-4612-4176-8
12. Sullivan, G. M., & Artino, A. R., Jr. (2013). Analyzing and interpreting data from Likert-type scales. Journal of Graduate Medical Education, 5(4), 541–542. doi: 10.4300/JGME-5-4-18
13. Taber, K. S. (2018). The use of Cronbach's alpha when developing and reporting research instruments in science education. Research in Science Education, 48, 1273–1296. doi: 10.1007/s11165-016-9602-2
14. Tall, D. (2004). Thinking through three worlds of mathematics. In M. J. Høines & A. B. Fuglestad (Eds.), Proceedings of the 28th Conference of the International Group for the Psychology of Mathematics Education (Vol. 4, pp. 281–288). PME.
15. The GAP Group. (2025). GAP – Groups, Algorithms, and Programming (Version 4.15.1) [Computer software]. GAP Project. https://www.gap-system.org
16. The SageMath Developers. (2026). SageMath [Computer software]. Zenodo. doi: 10.5281/zenodo.8042260