STATICALCS: AN INTERACTIVE WEB-BASED LEARNING TOOL WITH INTEGRATED CALCULATORS FOR STATICS OF RIGID BODIES

An Undergraduate Thesis
presented to the
Civil Engineering Department
College of Engineering
Mindanao State University
General Santos City

In partial fulfillment
of the requirements for the degree
Bachelor of Science in Civil Engineering

IAN CARL P. COÑA
SOPHIA DAPHNE C. FAELNAR

February 2026

Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Background of the Study

Engineering Mechanics is one of the fundamental components of engineering education, and Statics of Rigid Bodies serves as its essential foundation. The course develops the analytical skills needed to understand how forces and moments act on bodies at rest, making it a prerequisite for more advanced engineering subjects. In the Civil Engineering curriculum of Mindanao State University – General Santos (MSU–Gensan), *Statics of Rigid Bodies (ENS161)* is a core subject required for academic progression in the engineering program.

Despite its importance, many students find Statics difficult due to its mathematical requirements and abstract nature. Students often struggle with force systems, moments, and the construction of accurate *Free Body Diagrams (FBDs)*, which are necessary for solving Statics of Rigid Bodies problems. I. Salami and Perry found that learners commonly face conceptual and procedural difficulties in Statics, including challenges in visualizing forces and connecting equations to physical situations. These difficulties are further intensified by weak mathematical foundations, particularly in algebra and trigonometry, which are essential for problem-solving in Statics. Felix noted that incoming college students often exhibit declining mathematical readiness, affecting their ability to perform well in computation-heavy subjects.

Additionally, affective factors such as math anxiety contribute to students' performance challenges. Incierto et al. found that higher levels of math anxiety are associated with lower academic achievement, indicating that many students struggle not only with understanding concepts but also with the emotional stress associated with solving mathematical problems. Research also shows that appropriate tools, such as calculators and structured digital aids, can reduce computational errors and alleviate anxiety by allowing learners to focus on conceptual understanding (Segarra and Cabrera-Martínez).

Traditional lecture-based instruction, which relies heavily on verbal explanation and boardwork, often provides limited visualization and interactivity. Modern learners, particularly those belonging to Generation Z, respond more positively to technology-supported learning environments. Szymkowiak et al. emphasized that digital-native students benefit from online tools, multimedia resources, and interactive platforms that enhance engagement and comprehension.

Given these challenges, students benefit from supplementary learning tools that reinforce classroom instruction, enhance visualization, and support step-by-step problem solving. In response, the study developed *StatiCalcs*, a web-based interactive learning tool designed to help MSU–Gensan Civil Engineering students understand Statics of Rigid Bodies. The platform provides topic-aligned resources and integrated calculators that generate step-by-step solutions and visual representations, offering an accessible and technology-supported supplement to traditional instruction.

1.2 Statement of the Problem

This study aims to develop a web-based learning tool, *StatiCalcs*, specifically designed for the subject Statics of Rigid Bodies. After the development of the application, the study seeks to determine the level of perception of Civil Engineering students and instructors of MSU–General Santos regarding its usability, accessibility, and user satisfaction.

Specifically, this study aims to answer the following questions:

1. What is the level of perception of Bachelor of Science in Civil Engineering students of MSU–General Santos regarding *StatiCalcs* in terms of:

   1.1 Usability;

1.2 Accessibility; and

1.3 Satisfaction?

2. What is the level of perception of Bachelor of Science in Civil Engineering instructors of MSU–General Santos regarding *StatiCalcs* in terms of:

   2.1 Usability;

   2.2 Accessibility; and

   2.3 Satisfaction?

3. Is there a significant difference between the level of perception of Bachelor of Science in Civil Engineering students and instructors regarding *StatiCalcs* in terms of usability, accessibility, and satisfaction?

Null Hypothesis

Ho: There is no significant difference between the level of perception of Bachelor of Science in Civil Engineering students and instructors regarding *StatiCalcs* in terms of usability, accessibility, and satisfaction.

1.3 Scope and Limitations

This study will focus on the design, development, and evaluation of a web-based learning tool, *StatiCalcs*, intended to assist Bachelor of Science in Civil Engineering students of Mindanao State University – General Santos in studying the subject Statics of Rigid Bodies. The website will cover all major topics in Statics, including force systems, equilibrium, structures, centroids, moments of inertia, and friction. It will integrate step-by-step solutions and calculators to help users understand and solve problems interactively. The evaluation of the tool will focus on three aspects: usability, accessibility, and user satisfaction.

The respondents of the study will be limited to Bachelor of Science in Civil Engineering students, specifically second-year students currently enrolled in Statics of Rigid Bodies and students who have already taken the subject, as well as instructors teaching the same course in the College of Engineering of MSU–General Santos during the Academic Year 2025–2026. Data will be collected through survey questionnaires administered after respondents have explored and used the developed website.

The study will not measure students' academic performance or actual improvement in examination results. It will also exclude students and instructors from other engineering programs. Moreover, the study will focus solely on perceptions of the usability and effectiveness of the developed tool. It will not account for external factors such as internet connectivity, learning motivation, or respondents' prior knowledge.

The scope of this project will include the development of a website using modern web technologies, focusing on both frontend and backend functionality. The backend of the website will be built in Java, handling application logic and processing user interactions. The frontend interface will be designed with Tailwind CSS, ensuring a responsive, modern, and visually consistent user experience. Version control and collaboration will be managed through GitHub, while deployment and hosting will be automated via Netlify. ChatGPT will be utilized as a supplementary tool to assist in web design decisions, code arrangement, and debugging, providing guidance and improving workflow efficiency.

The project's delimitations and limitations will include reliance on specific technologies and platforms. The backend will be limited to Java, which may restrict integration with other programming languages or frameworks without additional configuration. The frontend design will depend on Tailwind CSS, and deviations from its utility-first approach may require additional custom CSS. While GitHub and Netlify will streamline version control and deployment, they will require internet connectivity and account setup, which could pose challenges in offline or restricted environments. Furthermore, because the website is hosted on a website hosting platform and not on a dedicated server-based environment, it cannot store or manage user data on the web. ChatGPT will serves only as a supportive tool and does not replace the developer's judgment or expertise. Additionally, the website's functionality and performance are constrained by the features provided by the chosen technologies and hosting platform, limiting scalability for highly complex or resource-intensive applications.

1.4 Significance of the Study

This study will focus on developing a web-based learning tool, *StatiCalcs*, to support the teaching and learning of Statics of Rigid Bodies among Bachelor of Science in Civil Engineering students at MSU–General Santos. The results and outcomes of this research may benefit the following:

Students. The developed website may serve as a supplementary learning tool, allowing students to practice problem-solving in Statics of Rigid Bodies through an interactive, user-friendly platform. As learners today are inclined toward technology-based education, *StatiCalcs* may provide an accessible and engaging way to enhance understanding and reinforce classroom instruction.

Teachers. The tool may be integrated into teaching strategies to improve the delivery of complex statics concepts. It may assist instructors in providing visual and computational demonstrations that complement lectures and classroom exercises.

School Administration. The study may provide the institution with an example of how to develop and implement digital learning tools to improve academic performance. It may also encourage future initiatives that promote technology-driven learning in other engineering courses.

Future Researchers. This study may serve as a reference for future studies aiming to develop similar educational tools or further improve *StatiCalcs* by adding new features, expanding its scope, or evaluating its long-term impact on student learning.

1.5 Theoretical Framework

Theoretical frameworks in quantitative research serve to provide a conceptual guide for choosing the concepts to be investigated, for suggesting research questions, and for framing the research findings (Corbin and Strauss). This study is anchored on the Technology Acceptance Model (TAM), developed by Davis, "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology"; "A technology acceptance model for empirically testing new end-user information systems," which explains how individuals accept and use technological systems. According to TAM, two primary constructs, Perceived Ease of Use (PEOU) and Perceived Usefulness (PU), shape a user's attitude toward a system, which subsequently influences satisfaction and intention to use the technology.

Perceived Ease of Use (PEOU) refers to the degree to which an individual believes that using a system will require minimal effort (Venkatesh and Davis). In this study, PEOU aligns with the usability dimension of *StatiCalcs*, reflecting how easily students and instructors can navigate the interface and perform tasks within the platform.

Perceived Usefulness (PU) is defined as the degree to which an individual believes that using a system will enhance their performance (Venkatesh and Davis). PU is used as the satisfaction variable in this study, representing users' perceptions of how effectively *StatiCalcs* supports their understanding and application of statics concepts.

Although TAM primarily emphasizes intention to use, this study adapts the model to evaluate usability, accessibility, and satisfaction, constructs essential for assessing a web-based learning tool. Accessibility is incorporated as an additional factor to capture the ease with which users can access and utilize the platform across various devices.

TAM provides valid, reliable, and easy to administer scales for the key constructs (Venkatesh and Davis), making it an appropriate theoretical basis for developing the questionnaire and interpreting differences in perceptions between students and instructors toward *StatiCalcs*.

1.6 Conceptual Framework

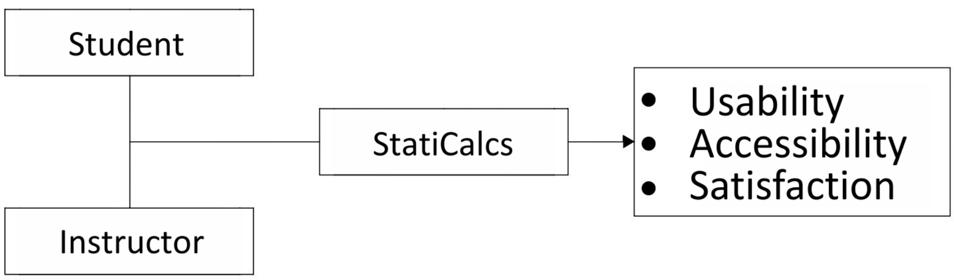

Figure 1.1: Conceptual Framework

This study is anchored in a conceptual framework that describes the relationship between users of the system and the web-based learning tool *StatiCalcs*, and how their interaction with the platform leads to the evaluation of three key system attributes: usability, accessibility, and satisfaction. The independent variables are the students and instructors, while the dependent variables are their level of perception in terms of usability, accessibility, and satisfaction.

StatiCalcs functions as the central intervention, offering computational features and step-by-step problem-solving support. The quality of users' interactions with the system is evaluated through three outcome variables: usability, accessibility, and satisfaction. Usability pertains to the efficiency and ease with which users navigate and perform tasks within the platform. Accessibility evaluates the system's ability to accommodate diverse users, devices, and learning conditions. Satisfaction measures the overall acceptance of the tool and the extent to which it meets user expectations.

Overall, the framework posits that students' and teachers' experiences with *StatiCalcs* directly shape their perceptions of its usability, accessibility, and satisfaction, thereby informing the system's effectiveness as a supplementary learning tool.

1.7 Definition of Terms

Accessibility
The extent to which Bachelor of Science in Civil Engineering students enrolled in Statics of Rigid Bodies and their instructors can easily access, navigate, and use the developed website, *StatiCalcs*, regardless of ability, to support effective learning and teaching.

Bachelor of Science in Civil Engineering Students
Refers to the respondents of the study who are currently enrolled in or have previously completed the course Statics of Rigid Bodies in the Academic Year 2025–2026. These participants will provide data regarding their perception of the usability, accessibility, and satisfaction of the developed website.

Bachelor of Science in Civil Engineering Instructors
Refers to the respondents of the study who are teaching the course Statics of Rigid Bodies during the Academic Year 2025–2026. Their feedback will contribute to evaluating the effectiveness, usability, and accessibility of the developed website as a supplementary learning tool.

Digital Natives	Individuals who grew up in the age of digital technology and are comfortable using computers, the Internet, and web-based applications. They represent the target population of this study.
Dynamics of Rigid Bodies	A branch of mechanics that examines the motion of non-deformable bodies under applied forces, including translational and rotational motion, using Newton's laws and principles of energy and momentum. Prerequisite: Statics of Rigid Bodies.
Free Body Diagram (FBD)	A Free Body Diagram (FBD) is a simplified graphical representation of a body or system isolated from its surroundings, showing all external forces and moments acting on it. It is used to visualize and solve problems involving equilibrium.
Mechanics of Deformable Bodies	Also known as the mechanics of materials, this field studies the behavior of solid bodies that deform under applied forces. It analyzes how stresses and strains are distributed within materials and relates these deformations to material properties. Prerequisite: Statics of Rigid Bodies.

Perception The way students and instructors interpret and evaluate their experiences with a learning tool, concept, or environment. In this study, it refers to how they perceive the usability, accessibility, and effectiveness of the developed website.

Statics of Rigid Bodies A branch of engineering mechanics that studies forces and their effects on rigid bodies, focusing on conditions of equilibrium where the sum of all forces and moments equals zero. This course is the main focus of the study.

Satisfaction The degree to which Bachelor of Science in Civil Engineering students enrolled in Statics of Rigid Bodies and their instructors feel content or fulfilled with the developed website, *StatiCalcs*, reflecting how well it meets their learning needs and expectations.

Usability The degree to which Bachelor of Science in Civil Engineering students enrolled in Statics of Rigid Bodies and their instructors can effectively, efficiently, and satisfactorily use the developed website, *StatiCalcs*, to achieve their learning and teaching goals.

Chapter 2
Review of Related Literature

The Review of Related Literature discusses previous studies and theories relevant to this research. It focuses on the challenges students face in learning Statics of Rigid Bodies, the use of web-based learning tools, and learners' perceptions of technology-assisted instruction. This review highlights the need for an interactive platform like *StatiCalcs* to support effective learning and teaching.

2.1 Statics of Rigid Bodies in Engineering Education

Statics of Rigid Bodies is a fundamental area of Engineering Mechanics concerned with understanding how forces and moments act on bodies that remain in equilibrium. As a foundational subject, Statics develops the analytical reasoning necessary for advanced topics such as Dynamics, Strength of Materials (Mechanics), Structural Analysis, and Fluid Mechanics.

In the Civil Engineering curriculum of Mindanao State University – General Santos (MSU-Gensan), as stated in MSU Board Resolution No. 375, s. 2017, Statics of Rigid Bodies (ENS161) is formally designated as a foundation course and a prerequisite for higher engineering subjects. Its role as a gateway subject underscores the need for students to develop strong problem-solving, visualization, and mathematical reasoning skills before moving on to more complex engineering analyses.

For Civil Engineering students, mastery of Statics is essential because it directly relates to designing structures that safely resist loads. Failure to grasp the principles of force systems, equilibrium, and moment calculations may hinder students' academic progression and reduce their preparedness for professional engineering practice (I. E. Salami et al.). Its foundational role in engineering education highlights the need for effective teaching approaches and supportive learning resources.

2.2 Conceptual Difficulties in Learning Statics

Learning Statics poses substantial conceptual challenges for students, mainly because the subject requires interpreting forces and interactions that are not directly observable. Students frequently struggle to visualize force directions, understand moment effects, and connect theoretical principles to real-world mechanical behavior. These conceptual gaps often lead to incorrect assumptions about how forces act on objects and how equilibrium conditions are achieved.

One of the most common sources of difficulty is the abstraction involved in translating a real object into a simplified engineering model. Students must isolate bodies, idealize supports, and mentally detach systems from their environment, steps essential to constructing valid Free Body Diagrams (FBDs). When learners are unable to visualize these relationships clearly, their problem-solving accuracy decreases significantly.

Misconceptions about force components, resultant forces, and moment calculations further contribute to students' struggles. Many learners also find it challenging to distinguish between couples, force systems, and distributed loads. These conceptual difficulties suggest the need for instructional approaches that enhance visualization, provide immediate feedback, and reduce students' dependence on rote memorization.

2.3 Procedural and Analytical Challenges

In addition to conceptual issues, I. Salami and Perry emphasized that students often face procedural difficulties when applying Statics principles to computational tasks. Many struggle to correctly break forces into components, apply equilibrium equations, or follow systematic steps when solving problems. These challenges frequently arise from gaps in prior mathematical knowledge, particularly in algebra and trigonometry.

Procedural errors also occur when students attempt to construct FBDs without fully understanding the physical systems involved. Incorrect or incomplete diagrams often lead to errors that persist through the entire solution process. According to I. Salami and Perry, procedural weaknesses combined with conceptual misunderstandings contribute to persistent difficulties in Statics, even among higher-year students.

Students also tend to rely on memorized formulas without understanding their derivations or applications. This approach hampers their ability to transfer knowledge to unfamiliar problem types or adapt to more complex engineering tasks. Such procedural weaknesses highlight the need for supplementary learning resources that guide students step by step and reinforce systematic problem-solving habits.

2.4 Mathematical Readiness in Statics

A strong mathematical foundation is a prerequisite for success in Statics of Rigid Bodies. However, many incoming engineering students lack proficiency in algebra, geometry, trigonometry, and vector operations, skills essential for analyzing forces and moments. Felix reports that declining math preparedness among first-year students negatively affects performance in computation-heavy engineering subjects.

Wenceslao conducted a study assessing the mathematical and analytical readiness of incoming first-year engineering students in Eastern Visayas, Philippines. The study revealed that 57% of students were not mathematically college-ready, demonstrating significant deficiencies in key areas such as calculus, trigonometry, and algebra, which are essential for problem-solving in engineering disciplines like Statics. Students struggled particularly with integration, derivatives, and trigonometric identities, indicating gaps in foundational skills needed to manipulate equations, solve simultaneous equations, resolve forces, and understand vector relationships. These weaknesses often lead to frustration, disengagement, and errors in calculating resultants, determining moment arms, and resolving angled forces.

The findings underscore the need for early interventions, remedial support, digital or supplementary learning tools that scaffold mathematical procedures, reducing the cognitive burden of complex computations and allowing learners to focus on conceptual understanding and practical application in Statics. Digital resources that automate or guide mathematical procedures may help students focus more on conceptual interpretation and problem comprehension.

2.5 Math Anxiety and Its Impact on Engineering Students

Math anxiety is a widely documented factor that affects students' performance in technical subjects. Incierto et al. found that students with higher levels of anxiety tend to perform poorly in mathematics-related courses, including engineering subjects that rely heavily on quantitative skills. This emotional barrier often causes learners to avoid practicing mathematical problems, ultimately reducing their mastery of essential skills.

In Statics, where problem-solving requires applying multiple mathematical steps, anxiety can impair students' ability to think clearly, interpret diagrams, and follow procedural logic. Reducing math anxiety is therefore an important aspect of improving student performance.

Several studies suggest that allowing students to use calculators, digital tools, or guided computational aids can mitigate anxiety by reducing the pressure associated with manual calculations. Segarra and Cabrera-Martínez found that structured calculator use improves students' confidence and accuracy, enabling them to concentrate more on conceptual understanding than on arithmetic manipulation.

2.6 Visualization and Spatial Ability in Engineering

Spatial reasoning is a key competency in engineering, particularly in Statics, where students must interpret diagrams, imagine 3D force interactions, and understand geometric relationships. Learners with weak spatial ability often struggle with vector representation, force decomposition, and moment visualization (Fontaine and Vallabh).

Visualization tools, including interactive diagrams and simulations, help develop these skills by allowing students to manipulate variables and observe changes in real time. Zeichner argues that simulations enhance student motivation and conceptual understanding, particularly in visually demanding subjects such as Statics. Given that Statics requires a combination of spatial and analytical skills, web-based learning tools that incorporate visualization are essential for supporting diverse learning needs.

De La Hoz et al. emphasized the effectiveness of visualization-based platforms in Statics learning through the integration of the spatial simulation tool Hapstatics. In their study, students interacted with worked examples while self-explaining each solution step, which improved conceptual understanding of static equilibrium. The researchers concluded that the integration of visualization tools is essential for students to achieve a deeper understanding of the physical behaviors illustrated in Statics exercises, as these environments help learners identify forces, reactions, and structural behavior more accurately.

They further noted that self-explanation practices significantly contribute to conceptual change, as students who verbalize and justify their solution steps demonstrate better problem-solving performance and error recognition. While this process may require prompting and structured guidance, interactive digital environments support these outcomes by encouraging explanation, reflection, and correction of misconceptions. This approach aligns with the purpose of *StatiCalcs*, which similarly provides visual and guided analytical support to strengthen conceptual understanding beyond text-based explanation.

2.7 Importance of Free Body Diagram Construction

The Free Body Diagram (FBD) is one of the most critical tools in Engineering Mechanics. It serves as the foundation for analyzing force interactions, solving equilibrium equations, and understanding mechanical behavior. Despite its importance, many students struggle to construct accurate FBDs, often omitting forces, misrepresenting directions, or misidentifying support reactions (I. Salami and Perry).

Errors in FBD construction typically result in incorrect equilibrium equations, making the entire solution process invalid. Researchers emphasize the need for learning tools that support FBD visualization, provide real-time diagrams, and reinforce correct diagramming techniques. Visual learning tools and graphical simulations have been shown to improve performance and deepen understanding in Statics.

2.8 Technology-Based Learning for Engineering Student

The concept of digital natives, introduced by Prensky, describes individuals who have grown up immersed in digital technology, such as computers, smartphones, and the internet. This group generally includes those born from the 1980s onward, commonly identified as Millennials, Generation Z, and Generation Alpha. These generations grew up during the rapid advancement of technology, which has profoundly influenced their lifestyles, behaviors, and learning preferences. Since they have never known a world without digital connectivity, technology has become an integral part of their identity and daily lives. According to Szymkowiak et al., educators should combine traditional teaching methods with modern, internet-based learning tools, including mobile applications and online videos, to accommodate the learning preferences. Zeichner similarly found that learning through simulation fosters a deeper understanding of abstract principles and concepts compared with conventional instruction. These findings are supported by

De La Hoz et al., who observed that traditional lecture-based teaching in statics often fails to address students' learning barriers and problem-solving difficulties.

Also, a comprehensive literature review by Tawafak et al. highlights that the continuous exposure of Gen Z to digital environments has transformed their learning preferences and expectations in higher education. Their familiarity with interactive, technology-driven systems opens opportunities for educators to design more engaging, adaptive digital learning platforms. Collectively, these studies emphasize the importance of integrating interactive, technology-enhanced approaches into higher education to meet the learning needs of digital natives in modern academic settings and to improve student engagement, comprehension, and analytical skills, especially in conceptually demanding courses such as Statics of Rigid Bodies.

Technology-based learning enhances engagement, improves retention, and provides flexibility for self-paced study. Web-based learning tools have been shown to promote active learning, encourage repeated practice, and support independent problem-solving. These tools can supplement traditional lectures, which often lack interactivity and visualization. Given these advantages, integrating technology into Statics instruction is essential to address students' difficulties and enhance their learning experience.

2.9 Digital Learning Tools and Student Engagement

Mobile learning has become widely used in education because it offers flexibility, accessibility, and opportunities for learners to complete academic tasks at their own pace. Research consistently shows that mobile learning applications enhance motivation and engagement by providing interactive features and allowing students to access materials anytime and anywhere. Pedraza and Canoy found that mobile apps increase student engagement by offering instant feedback and convenient access to tasks and resources, enabling learners to continue studying even outside formal class hours. Similarly, Hassan et al. emphasize that digital learning tools are most effective when they are simple to use, visually clear, and easy to navigate, especially for learners who struggle with complex instructions or text-heavy materials. These studies highlight that usability, accessibility, and clarity are fundamental qualities of effective digital learning environments.

Although the three studies focus on mobile applications, their findings are highly relevant to web-based platforms, as both fall under the broader category of digital learning tools. Mathur's dissertation supports this connection through the Technology Acceptance Model (TAM), which posits that users are more likely to adopt and continue using a digital tool when it is ea

2.10 Web-Based Learning in Engineering Education

Web-based learning has become a significant trend in modern education due to its accessibility, flexibility, and multimedia integration capabilities. Through internet-based platforms, students can access materials anytime, practice problems repeatedly, and receive immediate feedback. This is especially useful in technical subjects that require repeated exposure to problem-solving procedures.

Research by Abumandour on the potential of e-learning as an educational system for engineering topics highlighted the rapid technological advancements of the 21st century, which have transformed educational delivery methods. The researcher noted that advances in technology have led to the emergence of e-learning as a modern instructional approach widely adopted by educational institutions, public organizations, and academic libraries. The researcher highlighted that engineering education is increasingly moving toward a blended learning model that effectively integrates traditional "face-to-face" instruction with computer-assisted methodologies and internet-based learning. This hybrid approach enhances accessibility, flexibility, and learner engagement.

The study also outlined several challenges and obstacles that stakeholders, including teachers, professors, and librarians, must address to develop and sustain effective e-learning systems fully. These challenges include ensuring technological readiness, maintaining instructional quality, and supporting both instructors and students in adapting to digital learning environments. The researcher further proposed recommendations to strengthen the connection between e-learning and engineering education, thereby promoting a more adaptive, innovative, and inclusive academic experience. The researcher highlights the growing need for digital tools in engineering education as institutions shift toward blended and online learning systems.

In the study of Söllradl, which aims to enhance digital learning in Engineering Mechanics across several departments, the researcher developed an automated, scalable system to generate individualized assignments for students, supporting a more efficient, interactive approach to learning. To accomplish this, the researcher used a Python-based framework that automatically generates, distributes, and solves assignments for beam structures. Instead of relying solely on numerical solutions, the system emphasized symbolic computation using the *Euler–Bernoulli beam theory* to formulate and solve linear systems of equations. This method provided learners with a clearer understanding of the mechanical behavior of beam structures by reinforcing analytical and theoretical principles rather than focusing only on numerical outputs. Through this approach, this study demonstrated how digitalization and automation can be effectively integrated into engineering education to promote personalized learning, reduce instructors' manual workload, and enhance students' analytical and problem-solving skills.

Another significant advancement in web-based learning was introduced by Gfrerer et al., who developed an automated correction system for individualized exercise assignments in Engineering Mechanics courses. Their work addresses long-standing challenges in large classes, where manually creating and grading assignments limits the variety of problems students can encounter. With smaller problem pools, learners may replicate solutions without developing genuine conceptual understanding, and instructors may struggle to assess independent problem-solving skills.

To overcome these issues, the researchers developed a scalable digital framework capable of generating, distributing, and automatically correcting personalized exercises in Statics, Strength of Materials, Dynamics, and Hydrostatics. Each student receives a unique set of problems, promoting academic integrity and encouraging active, self-directed learning. A quantitative survey among Statics of Rigid Bodies students showed strong acceptance of the tool, with many reporting enhanced reflective learning and improved engagement. The authors concluded that automated correction systems significantly improve efficiency in mechanics education by reducing instructor workload while supporting deeper learner understanding.

Web-based tools allow for unified access across devices, making them ideal supplements for engineering subjects that require visual illustrations and computational support. These features align perfectly with the needs of Statics students who benefit from guided examples, diagrams, and automated computations.

2.11 Related Systems and Existing Educational Tools

Several digital tools and online platforms have been developed to support learning in mathematics and engineering, but each has limitations when applied explicitly to Statics of Rigid Bodies. Symbolab, for instance, provides automated, step-by-step solutions for algebra, calculus, and trigonometric problems. A study by Paulin and Jean Baptiste aimed to determine the effectiveness of the Symbolab Calculator in improving second-year students' ability to solve trigonometric equations. The findings suggest that Symbolab can enhance students' problem-solving skills and understanding of mathematical concepts. However, despite these benefits, Symbolab lacks dedicated features for statics topics such as equilibrium, force systems, and truss structures, limiting its direct applicability in engineering Statics courses.

GeoGebra offers dynamic and interactive visualizations for geometry and algebra. Its visual approach enhances students' spatial reasoning and understanding of mathematical relationships. However, GeoGebra does not include built-in tools for engineering Statics, such as truss analysis or 2D/3D equilibrium calculations.

More specialized platforms, such as SkyCiv, provide advanced structural analysis tools for 2D and 3D frames and trusses. While highly useful for engineering applications, many features require paid subscriptions, reducing accessibility for undergraduate students.

Other digital innovations, such as the automated correction systems developed by Gfrerer et al., focus on generating individualized problem sets and automated grading in Engineering Mechanics courses. These systems enhance self-directed learning but primarily support assessment rather than conceptual understanding. Similarly, symbolic computation tools developed by Söllradl emphasize automated and symbolic solutions for beam structures in Strength of Materials, but do not provide comprehensive coverage of fundamental statics topics.

Although these tools contribute significantly to engineering education, none provide an integrated, free, and Statics-specific learning environment that combines step-by-step solutions, real-time computations, and dynamic visualizations. This gap underscores the need for a specialized platform like *StatiCalcs*, designed for undergraduate Statics students, providing an accessible, interactive, and comprehensive learning tool that addresses both conceptual understanding and practical problem-solving.

2.12 Summary of Literature

The literature consistently shows that Statics is conceptually demanding, procedurally complex, and dependent on strong mathematical foundations. Students struggle with visualization, FBD construction, force interactions, and equilibrium analysis. Traditional instruction alone often fails to address these learning gaps.

Research also emphasizes the importance of digital tools, visual learning, and web-based platforms in enhancing student understanding, especially for digital-native learners. Existing tools such as Symbolab, GeoGebra, and SkyCiv offer partial support but do not provide a comprehensive Statics learning environment.

The reviewed literature clearly indicates the need for an accessible, interactive, and specialized web-based tool to support students' learning in Statics. This provides strong justification for developing *StatiCalcs*, which integrates visualization, computation, and conceptual understanding to address the gaps identified in previous studies.

Chapter 3
Methodology

This chapter presented the research method, respondents, locale, instrument, and procedure used in the study. It explained how the research was conducted to determine the perceptions of Bachelor of Science in Civil Engineering students and instructors toward *StatiCalcs*. This interactive web-based learning tool was developed for the Statics of Rigid Bodies course.

3.1 Research Method

This study used a quantitative research method, which focused on collecting and analyzing numerical data to describe patterns, relationships, and differences among variables. According to Apuke, quantitative research involved a systematic investigation that used statistical techniques to produce objective and measurable results. This method was appropriate for the present study because it sought to gather data from Civil Engineering students and instructors to assess and compare their perceptions of the developed web-based learning tool, *StatiCalcs*.

Specifically, a correlational quantitative approach was applied. As defined by Creswell, this approach examined the relationship or difference between two or more variables to determine whether a significant association existed. In this study, it was used to determine the significant difference between the perceptions of students and instructors regarding the usability, accessibility, and satisfaction of *StatiCalcs*. This method allowed the researchers to statistically analyze the degree to which these two groups differed in their perceptions, providing insights into the effectiveness and acceptance of the developed learning tool across users.

3.2 Research Respondents

The respondents of this study were selected Bachelor of Science in Civil Engineering students and teaching faculty from the College of Engineering at Mindanao State University – General Santos (MSU–Gensan). The student respondents included second-year Bachelor of Science in Civil Engineering students who were either currently enrolled in or had previously taken the course Statics of Rigid Bodies (ENS161). This selection ensured that participants had sufficient background and experience with the subject to provide reliable feedback on the developed web-based learning tool, *StatiCalcs*.

The faculty respondents, on the other hand, consisted of instructors who were currently teaching or had previously taught Statics of Rigid Bodies, as they could provide expert evaluation regarding the tool's usability, accessibility, and relevance to the course content.

Purposive sampling was used to select students who had taken or were currently enrolled in Statics of Rigid Bodies, and instructors who had experience teaching the course.

3.3 Research Locale

This study was conducted at the College of Engineering, Mindanao State University – General Santos (MSU–Gensan), located in Fatima, General Santos City. The college offered various engineering programs, including Civil, Mechanical, Electrical, and Agricultural and Biosystems Engineering. The research primarily focused on the H Building, where most classroom lectures, computer laboratory sessions, and faculty offices were situated. This location was ideal for the study since it housed both the student respondents taking Statics of Rigid Bodies and the faculty members teaching the subject.

3.4 Research Procedure

The researchers conceptualized, designed, and developed a web-based learning tool called *StatiCalcs*, an interactive platform with integrated calculators specifically designed for the subject Statics of Rigid Bodies. The system underwent evaluation and approval by experts in the field to ensure its accuracy, functionality, and relevance to the course.

A researcher-made questionnaire was developed to measure the level of perception of Civil Engineering students and instructors in terms of usability, accessibility, and satisfaction. This questionnaire was then evaluated by academic experts in the field to ensure its validity and reliability before being used in the study.

Before data gathering, a formal letter of permission to conduct the study was submitted to the Dean of the College of Engineering at Mindanao State University–General Santos. Respondents were first allowed to explore and use the website *StatiCalcs* before the data collection process. They were given approximately fifteen (15) days to use and interact with the platform to ensure adequate exposure and familiarity with its features and tools before answering the survey. The researchers administered the questionnaire to the selected respondents, composed of Bachelor of Science in Civil Engineering students and instructors. After data collection, the gathered responses were encoded, organized, and subjected to statistical analysis.

Participation in the study was voluntary, and all responses were treated with strict confidentiality to ensure the privacy and protection of the respondents.

3.4.1 Website Development Process

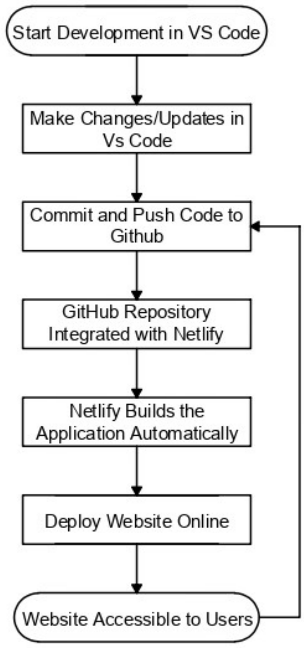

Figure 3.1: System Flow Chart

The website was developed using a modern workflow, beginning with the selection of Java as the backend language due to its reliability, security, and strong support for building large, complex web applications. GitHub was used as the code repository to manage versions, collaborate, and integrate with deployment tools. For the frontend, Tailwind CSS was chosen to create a clean, responsive, and consistent user interface using its utility-first styling approach. ChatGPT assisted throughout the development process by helping with code structuring, debugging, and design decisions, improving efficiency and workflow. Netlify was used as the hosting and deployment platform. By connecting the GitHub repository to Netlify, every update pushed to GitHub was automatically built and deployed online, ensuring continuous delivery and easy accessibility of the website to users.

3.4.2 Website Navigation and Functional Flow

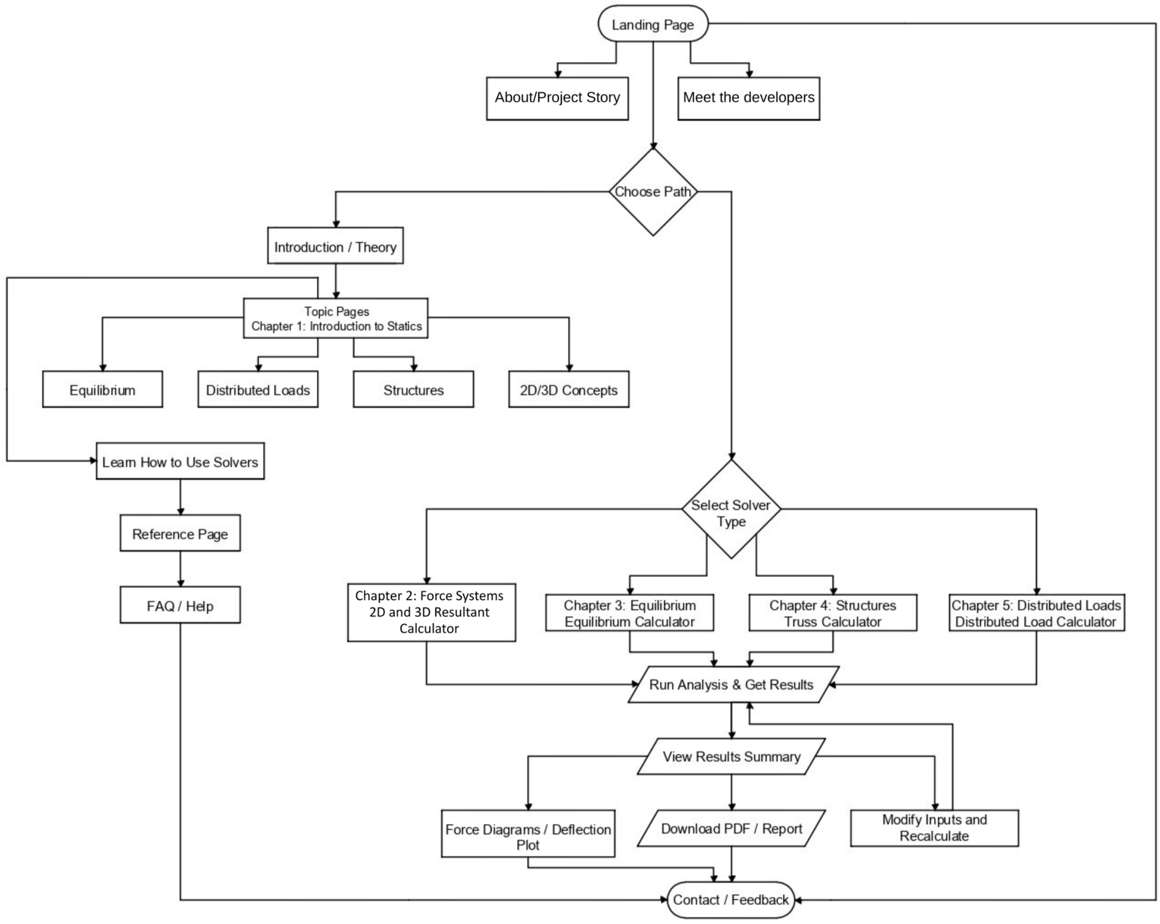

Figure 3.2: User journey flow chart

The website began at the Landing Page, which acted as the main entry point for users. From this page, users were provided with navigational options to explore introductory sections such as About / Project Story, Meet the Developers, and Contact / Feedback. These sections helped users understand the objectives of the project, know the creators behind the system, and provide a communication channel for inquiries or suggestions.

From the Landing Page, users encountered a decision point labeled Choose Path, where they decided whether to proceed to the Learning Section or directly access the Solver Tools.

If the user selected the Learning Path, they were directed to the Introduction /

Theory page, which presented foundational concepts in statics. Following this, users proceeded to Topic Pages (Chapter 1: Introduction to Statics), which included lessons on essential topics such as Equilibrium, Distributed Loads, Structures, and 2D/3D Concepts. After gaining theoretical knowledge, users moved to the Learn How to Use Solvers section, which offered instructional guides on how to properly use the computational tools for structural analysis.

To further support learning, users visited the Reference Page, which contained formula compilations, example problems, and relevant learning materials. An FAQ / Help section was also available to address common issues and user concerns. If additional support was needed, users contacted the developers directly through the Contact Page.

Alternatively, if the user chose the direct problem-solving pathway, they proceeded to a second decision point called Select Solver Type. The available solver tools were aligned with the chapters covered in the theoretical section, including:

- Chapter 1: Introduction to Statics
- Chapter 2: Force Systems — 2D and 3D Resultant Force Calculator
- Chapter 3: Equilibrium — Equilibrium Calculator
- Chapter 4: Structures — Truss Analysis Calculator
- Chapter 5: Distributed Loads — Structural Analysis Calculator for Distributed Load Effects

After selecting a specific solver, the user proceeded to Run Analysis and Get Results. After the computation, a View Results Summary page displayed the output in a clear, structured format.

From this results page, users had three options:

1. Generate and view Force Diagrams / Deflection Plot

2. Download results as PDF or formatted report
3. Modify input parameters and Recalculate to explore different solutions or optimize results

This structured flow allows users to either learn fundamental engineering concepts before solving problems or directly perform structural analysis based on their needs. The design ensures flexibility, user engagement, and accessibility while supporting both educational and analytical purposes.

3.4.3 Website System Architecture

The *Components* folder contained reusable elements of the website, specifically the Header and Footer, which helped maintain consistent navigation and accessibility across all pages. The *Header* file inside this folder included the main navigation menu with the *Home* button that redirected users to the homepage, along with a *Topics* dropdown that provided quick access to various chapters and calculators such as:
Chapter 1: Introduction to Statics, *Chapter 2: Force Systems with a 2D Resultant Solver*, *Chapter 3: Equilibrium with an Equilibrium Solver*, *Chapter 4: Structures with a Truss Calculator*, and *Chapter 5: Distributed Loads*. This dropdown was designed to make browsing easier and more efficient for users.

The header also contained the About page that presented the purpose and background of the website. Meanwhile, the Footer file contained additional navigation links including *About*, *References*, *Contact*, and *Developers*. The About button directed users to the website's About page; the References page provided the sources used in developing the calculators and learning materials; the Contact page allowed users to reach the proponents of the thesis; and the Developers page displayed information about the creators of the website. These components together ensured that users could easily access important sections and navigate the site smoothly.

The *Calcs* folder contained the core Java code files that powered the calculators integrated into the website. These files were responsible for performing the computational logic, processing user inputs, and generating the corresponding solutions based on engineering principles and formulas. Each calculator, such as those for force systems, equilibrium, truss analysis, and distributed loads, relied on the scripts stored in this folder to accurately compute results. This made the Calcs folder an essential component of the website's functionality, as it handled the main calculation processes that supported the learning and problem-solving features of the platform.

The *App* folder contained all the front-end code for all website pages, styled using Tailwind CSS, which controlled their visual layout, responsiveness, and overall user interface. Each page inside the App folder was responsible for either displaying calculators, such as for 2D Resultant, 3D Solver, Equilibrium, Distributed Loads, and Structural Analysis, or for providing supporting information, such as Introduction, References, About, Contact, and Developer details. The Tailwind CSS classes ensured that all these pages were consistently designed, user-friendly, and visually appealing, making navigation smooth and calculator usage clear and efficient. In summary, the App folder managed the appearance, layout, and behavior of both the calculator and informational pages from the user's perspective.

3.5 Statistical Tools

A researcher-made questionnaire was utilized in this study. It consisted of three parts, each measured on a five-point Likert scale, to evaluate respondents' perceptions of the developed web-based learning tool, *StatiCalcs*, specifically in terms of usability, accessibility, and user satisfaction.

The items were adapted from previous studies and guided by the Technology Acceptance Model (TAM). Usability items were adapted from Mathur and TAM ease-of-use constructs, accessibility items were adapted from Pedraza and Canoy and general e-learning accessibility indicators, and satisfaction items were adapted from both Pedraza and Canoy and TAM usefulness constructs. All items were modified to reflect the specific context of the *StatiCalcs* platform.

Each part of the instrument consisted of 10 items, resulting in a total of 30 items. Respondents rated each statement on a scale from 1 (Strongly Disagree) to 5 (Strongly Agree), allowing for quantitative analysis of their perceptions regarding the usability, accessibility, and satisfaction of *StatiCalcs*.

To ensure the internal consistency and reliability of the researcher-made instrument, Cronbach's alpha was computed for each of the three sections: usability, accessibility, and satisfaction. The reliability testing confirmed that the questionnaire items were statistically consistent and appropriate for use in the study.

Table 3.1: Likert Scale for Respondent Perception

Scale	**Range**	**Description**	**Interpretation**
5	4.21 – 5.00	Strongly Agree	The respondent fully agrees with the statement.
4	3.41 – 4.20	Agree	The respondent generally agrees with the statement.
3	2.61 – 3.40	Neutral	The respondent neither agrees nor disagrees with the statement.
2	1.81 – 2.60	Disagree	The respondent generally disagrees with the statement.
1	1.00 – 1.80	Strongly Disagree	The respondent fully disagrees with the statement.

3.6 Statistical Treatment

This research utilized weighted mean and frequency distribution to determine the level of perception of the respondents toward the developed web-based learning tool, *StatiCalcs*, in terms of usability, accessibility, and user satisfaction. Each item in the questionnaire was measured using a five-point Likert scale, where respondents indicated their level of agreement or perception. The weighted mean was computed to quantify the overall perception of students and instructors, providing a clear and interpretable measure of how the tool was perceived across its different aspects. Frequency distribution was also used to show how many respondents selected each response, offering a detailed view of the spread of opinions and highlighting areas of agreement or disagreement.

In addition, an independent samples t-test was conducted to determine whether there was a significant difference between the perceptions of students and instructors regarding the usability, accessibility, and satisfaction of the developed learning tool. By combining these statistical treatments, the study provided both a summarized and comprehensive understanding of respondents' perceptions.

References

Abumandour, El-Shaimaa Talaat. "Applying e-learning system for engineering education – challenges and obstacles." *Journal of Research in Innovative Teaching & Learning*, vol. 15, no. 2, 6 Oct. 2021, pp. 150–69. https://doi.org/10.1108/JRIT-06-2021-0048.

Apuke, Oberiri. "Quantitative Research Methods : A Synopsis Approach." *Arabian Journal of Business and Management Review (kuwait Chapter).*, vol. 6, 1 Oct. 2017, pp. 40–47. https://doi.org/10.12816/0040336.

Corbin, Juliet, and Anselm Strauss. *Basics of Qualitative Research (3rd ed.): Techniques and Procedures for Developing Grounded Theory.* SAGE Publications, 2008, https://doi.org/10.4135/9781452230153.

Davis, Fred D. "A technology acceptance model for empirically testing new end-user information systems : theory and results." 1985. Massachusetts Institute of Technology, Thesis. dspace.mit.edu/handle/1721.1/15192. Accessed 9 Dec. 2025.

———. "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology." *MIS Quarterly*, vol. 13, no. 3, 1989, Publisher: Management Information Systems Research Center, University of Minnesota, pp. 319–40. *JSTOR*, https://doi.org/10.2307/249008.

De La Hoz, Jose L., et al. "Promoting Students' Conceptual Change in Statics through Self-Explanation Strategy in a Remote Learning Context." *9th Research in Engineering Education Symposium (REES 2021) and 32nd Australasian Association for Engineering Education Conference (REES AAEE 2021).* Proc. of 9th Research in Engineering Education Symposium & 32nd Australasian Association for Engineering Education Conference, Research in Enineering Education Network (REEN), 2022, pp. 215–23, https://doi.org/10.52202/066488-0024.

De La Hoz, Jose Luis, et al. "Self-explanation activities in statics: A knowledge-building activity to promote conceptual change." *Journal of Engineering Education*, vol. 112, no. 3, 2023, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jee.20531, pp. 741–68. https://doi.org/10.1002/jee.20531.

Felix, Criselda S. “Analyzing the Impact of Declining Math Skills on College Readiness for Incoming Freshmen.” Vol. 7, no. 1, 2023.

Fontaine, Maxine, and Chaitanya Vallabh. “Correlating Common Errors in Statics Problem Solving with Spatial Ability.” *2024 ASEE Annual Conference & Exposition Proceedings*. Proc. of 2024 ASEE Annual Conference & Exposition, ASEE Conferences, June 2024, p. 47079, https://doi.org/10.18260/1-2--47079.

Gfrerer, Michael H., et al. “Teaching mechanics with individual exercise assignments and automated correction.” *PAMM*, vol. 23, no. 3, Nov. 2023, e202300168. *arXiv*, arxiv.org/abs/2308.12694[physics], https://doi.org/10.1002/pamm.202300168.

Hassan, Jawad Ul, et al. “USABILITY EVALUATION OF MOBILE LEARNING APPS FOR SLOW LEARNERS.” 10 Nov. 2022. tianjindaxuexuebao.com/details.php?id= DOI:10.17605/OSF.IO/URSNJ. Accessed 9 Dec. 2025.

Incierto, Theresa Christine, et al. “Mathematics Anxiety Level of The First Year Education Students in Manduae City College in The New Normal in Relation To Their Academic Performance \textbar PDF \textbar Correlation And Dependence \textbar Anxiety.” PhD dissertation. www.scribd.com/document/516791046/Mathematics-Anxiety-Level-of-the-First-Year-Education-Students-in-Manduae-City-College-in-the-New-Normal-in-Relation-to-Their-Academic-Performance-1. Accessed 22 Oct. 2025.

Mathur, Roopa. “Students’ Perceptions of a Mobile Application for College Course Management Systems.” *Walden Dissertations and Doctoral Studies*, 1 Jan. 2011. scholarworks.waldenu.edu/dissertations/925.

Paulin, Manirakiza, and Ndagijimana Jean Baptiste. “Effectiveness of the Symbolab Calculator in Improving Second Year Science and Mathematics Students Ability to

Solve Trigonometric Equations." *EAST AFRICAN JOURNAL OF EDUCATION AND SOCIAL SCIENCES*, vol. 4, no. 6, 1 June 2024, pp. 29–38. https://doi.org/10.46606/eajess2023v04i06.0331.

Pedraza, Salvador Miguel, and Orbel Canoy. "Assessing the Effectiveness of Mobile Learning Apps in Enhancing Student Learning and Engagement in Selected Secondary Public Schools in General Trias City." *Psychology and Education: A Multidisciplinary Journal*, vol. 45, 2025. https://doi.org/10.70838/pemj.450909.

Salami, Ibukunoluwa, and Logan Perry. "Challenges in Engineering Statics: Students' Perceptions of Their Difficulties." *2025 ASEE Annual Conference & Exposition Proceedings*. Proc. of 2025 ASEE Annual Conference & Exposition, ASEE Conferences, June 2025, p. 56077, https://doi.org/10.18260/1-2--56077.

Salami, Ibukunoluwa E., et al. "Exploring High DFW Rates in an Engineering Statics Course: Insights from Faculty and Teaching Assistants." *2024 ASEE Midwest Section Conference Proceedings*. Proc. of 2024 ASEE Midwest Section Conference, ASEE Conferences, Sept. 2024, p. 49390, https://doi.org/10.18260/1-2-115.1139-49390.

Segarra, Jaime, and Abel Cabrera-Martínez. "Does the use of the calculator reduce anxiety in the study of differential and integral calculus?" *European Journal of Science and Mathematics Education*, vol. 12, no. 1, 1 Jan. 2024, pp. 97–111. https://doi.org/10.30935/scimath/13867.

Söllradl, Sophie. "Symbolic calculation of beam structures."

Szymkowiak, Andrzej, et al. "Information technology and Gen Z: The role of teachers, the internet, and technology in the education of young people." *Technology in Society*, vol. 65, 1 May 2021, p. 101565. https://doi.org/10.1016/j.techsoc.2021.101565.

Tawafak, Ragad M, et al. "Innovative Smart Phone Learning System for Graphical Systems within COVID-19 Pandemic." *Contemporary Educational Technology*, vol. 13, no. 3, 2 May 2021, ep306. https://doi.org/10.30935/cedtech/10879.

Venkatesh, Viswanath, and Fred D. Davis. "(PDF) A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies." *ResearchGate*, 6 Aug. 2025. https://doi.org/10.1287/mnsc.46.2.186.11926.

Wenceslao, Perante. "Mathematical Readiness of Freshmen Engineering Students (K-12 2020 Graduates) in Eastern Visayas in the Philippines." *Asian Journal of University Education*, vol. 18, no. 1, 14 Feb. 2022, p. 191. https://doi.org/10.24191/ajue.v18i1.17187.

Zeichner, Orit. "Using Simulations to Improve Achievements and Motivation in ICT Studies." *Journal on School Educational Technology*, vol. 16, no. 1, 2020, Publisher: i-manager Publications ERIC Number: EJ1354679, pp. 11–24. https://doi.org/10.26634/jsch.16.1.17426.

Appendix A

Research Questionnaire

Instructions

Please read each statement carefully. Put a check mark on the option that best describes your perception of the developed web-based learning tool, **StatiCalcs**. There are no right or wrong answers. All responses will be treated with strict confidentiality.

Part I. Respondent Information

Name (Optional): ________________________________

Status (Please check one): ☐ Student ☐ Instructor

Year Level (For students only): ________________________________

Have you used the StatiCalcs website? ☐ Yes ☐ No

Part II. Usability

This section evaluates how easy and efficient the StatiCalcs platform is to use.

Scale	Range	Description	Interpretation
5	4.21 – 5.00	Strongly Agree	The respondent fully agrees with the statement.
4	3.41 – 4.20	Agree	The respondent generally agrees with the statement.
3	2.61 – 3.40	Neutral	The respondent neither agrees nor disagrees with the statement.
2	1.81 – 2.60	Disagree	The respondent generally disagrees with the statement.
1	1.00 – 1.80	Strongly Disagree	The respondent fully disagrees with the statement.

No.	Statement	5	4	3	2	1
1	I find StatiCalcs easy to navigate.	☐	☐	☐	☐	☐
2	The functions of the platform are understandable.	☐	☐	☐	☐	☐
3	The interface is user-friendly.	☐	☐	☐	☐	☐
4	The instructions within the platform are clear.	☐	☐	☐	☐	☐
5	I can use the platform without guidance from others.	☐	☐	☐	☐	☐
6	The organization of features makes the tool convenient to use.	☐	☐	☐	☐	☐
7	I learn to use the tool quickly.	☐	☐	☐	☐	☐
8	The platform reduces the effort needed to perform Statics computations.	☐	☐	☐	☐	☐
9	The platform is responsive when performing calculations.	☐	☐	☐	☐	☐
10	Overall, the platform is easy to use.	☐	☐	☐	☐	☐

Part III. Accessibility

This section evaluates how easily the StatiCalcs platform can be accessed.

No.	Statement	5	4	3	2	1
1	The platform is accessible across multiple devices.	☐	☐	☐	☐	☐
2	I can use the platform without installation issues.	☐	☐	☐	☐	☐
3	The platform loads properly on my device.	☐	☐	☐	☐	☐
4	I can access the platform with stable performance.	☐	☐	☐	☐	☐
5	The platform is available whenever I need it.	☐	☐	☐	☐	☐
6	The features are accessible with minimal internet requirement.	☐	☐	☐	☐	☐
7	The text and interface elements are readable.	☐	☐	☐	☐	☐
8	The platform works properly on commonly used browsers.	☐	☐	☐	☐	☐
9	I can access the platform without technical difficulties.	☐	☐	☐	☐	☐
10	Overall, the platform is accessible and convenient to open.	☐	☐	☐	☐	☐

Part IV. User Satisfaction

This section evaluates overall satisfaction with the StatiCalcs platform.

No.	Statement	5	4	3	2	1
1	I am satisfied with the overall performance of the platform.	☐	☐	☐	☐	☐
2	The computation results provided are accurate.	☐	☐	☐	☐	☐
3	The platform meets my expectations as a Statics tool.	☐	☐	☐	☐	☐
4	I find the platform helpful for learning and solving Statics problems.	☐	☐	☐	☐	☐
5	The platform improves my efficiency in working on Statics tasks.	☐	☐	☐	☐	☐
6	I am satisfied with the speed of calculations.	☐	☐	☐	☐	☐
7	I am satisfied with the layout and design.	☐	☐	☐	☐	☐
8	I would recommend the platform to other students or instructors.	☐	☐	☐	☐	☐
9	The platform enhances my understanding of Statics.	☐	☐	☐	☐	☐
10	Overall, I am satisfied with the StatiCalcs platform.	☐	☐	☐	☐	☐

Appendix B

Source Code

Front End

```
import Link from "next/link";
import Header from "../components/Header";
import Footer from "../components/Footer";


export default function HomePage() {
  return (
    <main className="bg-gray-50 min-h-screen flex flex-col">
      <Header />


      {/* Landing Page*/}
      <section className="flex flex-col items-center text-center
py-12 bg-gray-50 flex-grow">
        <h1 className="text-[20px] md:text-[80px] font-bold mb-2
text-black">
          Stati
          <span className="text-[#1848a0]">
            Calcs
          </span>
        </h1>
        <p className="text-gray-600 mb-12 text-[18px]">
          Interactive calculators for learning and solving
Statics of Rigid Bodies.
        </p>


        {/* Chapter List */}
        <div className="w-full max-w-2xl">
          <div className="grid grid-cols-2 gap-4 items-center">

            {/* Chapter 1 */}
            <p className="text-left text-black text-[18px]
font-bold font-bold">Chapter 1: Introduction to Statics</p>

            <Link
              href="/Introduction"
              className="bg-[#1848a0] text-white px-6 py-3
rounded-md shadow hover:bg-[#163d8a] transition text-[18px]"
            >
              Introduction
            </Link>
```

```
            {/* Chapter 2 */}
            <p className="text-left text-black text-[18px]
font-bold font-bold">Chapter 2: Force Systems</p>
            <Link
              href="/2D-solver"
              className="bg-[#1848a0] text-white px-6 py-3
rounded-md shadow hover:bg-[#163d8a] transition text-[18px]"
            >
              2D Resultant Solver
            </Link>


            {/* Chapter 3 */}
            <p className="text-left text-black text-[18px]
font-bold font-bold">Chapter 3: Equilibrium</p>
            <Link
              href="/Equilibrium"
              className="bg-[#1848a0] text-white px-6 py-3
rounded-md shadow hover:bg-[#163d8a] transition text-[18px]"
            >
              Equilibrium Solver
            </Link>


            {/* Chapter 4 */}
            <p className="text-left text-black text-[18px]
font-bold font-bold font-bold">Chapter 4: Structures</p>
            <Link
              href="/Structures"
              className="bg-[#1848a0] text-white px-6 py-3
rounded-md shadow hover:bg-[#163d8a] transition text-[18px]"
            >
              Truss Calculator
            </Link>


            {/* Chapter 5 */}
            <p className="text-left text-black text-[18px]
font-bold font-bold font-bold">Chapter 5: Distributed Loads</p>
            <Link
              href="/Distributed-Loads"
              className="bg-[#1848a0] text-white px-6 py-3
rounded-md shadow hover:bg-[#163d8a] transition text-[18px]"
            >
              Structures Solver
            </Link>


            {/* Chapter 6 */}
            <p className="text-left text-black text-[18px]
font-bold font-bold font-bold">Chapter 6: Friction</p>
```

```
            <button className="border-2 border-[#1848a0]
text-[#1848a0] px-6 py-3 rounded-md hover:bg-[#163d8a]
hover:text-white transition text-[18px]">
              Coming Soon
            </button>


            {/* Chapter 7 */}
            <p className="text-left text-black text-[18px]
font-bold font-bold font-bold">Chapter 7: Virtual Work</p>
            <button className="border-2 border-[#1848a0]
text-[#1848a0] px-6 py-3 rounded-md hover:bg-[#163d8a]
hover:text-white transition text-[18px]">
              Coming Soon
            </button>


          </div>
        </div>
      </section>


      <Footer />
    </main>
  );
}
```

2D Solver Code

```
"use client";


import { useRef, useState } from "react";
import Header from "../../components/Header";
import Footer from "../../components/Footer";
import "katex/dist/katex.min.css";
import { BlockMath } from "react-katex";


/* Force System Logic */
class ForceSystem2D {
  vectors: { fx: number; fy: number; magnitude: number; angleDeg:
number }[];


  constructor() {
    this.vectors = [];
  }


  addForce(magnitude: number, angleDeg: number) {
    const angleRad = (angleDeg * Math.PI) / 180;
    const fx = magnitude * Math.cos(angleRad);
    const fy = magnitude * Math.sin(angleRad);
    this.vectors.push({ fx, fy, magnitude, angleDeg });
  }


  stepByStepSolution() {
    const steps: string[] = [];
    steps.push("Step 1: Resolve each force into components:");


    let sumFx = 0;
    let sumFy = 0;


    this.vectors.forEach((v, i) => {
      steps.push(
        `\\text{Force ${i + 1}: }
|F|=${v.magnitude}\\,\\text{kN},\\; \\theta=${v.angleDeg}^\\circ`
      );


      steps.push(`
        \\begin{align*}
        F_{x${i + 1}} &=
${v.magnitude}\\cos(${v.angleDeg}^\\circ) \\\\
```

```
                      &= ${v.fx.toFixed(3)}\\,\\text{kN} \\\\
        F_{y${i + 1}} &=
${v.magnitude}\\sin(${v.angleDeg}^\\circ) \\\\
                      &= ${v.fy.toFixed(3)}\\,\\text{kN}
        \\end{align*}
      `);


      sumFx += v.fx;
      sumFy += v.fy;
    });


    steps.push("Step 2: Sum of components:");
    steps.push(`
      \\begin{align*}
      \\Sigma F_x &= ${sumFx.toFixed(3)}\\,\\text{kN} \\\\
      \\Sigma F_y &= ${sumFy.toFixed(3)}\\,\\text{kN}
      \\end{align*}
    `);


    const R = Math.hypot(sumFx, sumFy);
    const theta = (Math.atan2(sumFy, sumFx) * 180) / Math.PI;


    const arrow = theta >= 0 ? "↺" : "↻";


    steps.push("Step 3: Resultant force:");
    steps.push(`
      \\begin{align*}
      R &= \\sqrt{(\\Sigma F_x)^2 + (\\Sigma F_y)^2} \\\\
        &= ${R.toFixed(3)}\\,\\text{kN} \\\\
      \\theta &= \\tan^{-1}\\left(\\tfrac{\\Sigma F_y}{\\Sigma
F_x}\\right) \\\\
              &= ${theta.toFixed(2)}^\\circ
${arrow}\\,\\text{from +x axis}
      \\end{align*}
    `);


    return { steps, sumFx, sumFy, R, theta };
  }
}


type ForceInput = {
  magnitude: string;
  angle: string;
};
```

```
type ForceResult = {
  steps: string[];
  sumFx: number;
  sumFy: number;
  R: number;
  theta: number;
};


/*FULL FBD FOR STEP 4 (all forces + resultant)*/
function ResultantFBD({
  forces,
  result,
}: {
  forces: ForceInput[];
  result: ForceResult;
}) {
  const vectors = forces
    .map((f) => {
      const m = parseFloat(f.magnitude);
      const a = parseFloat(f.angle);
      if (isNaN(m) || isNaN(a)) return null;
      const rad = (a * Math.PI) / 180;
      return { x: m * Math.cos(rad), y: m * Math.sin(rad) };
    })
    .filter(Boolean) as { x: number; y: number }[];


  const R = { x: result.sumFx, y: result.sumFy };


  const magnitudes = [
    ...vectors.map((v) => Math.hypot(v.x, v.y)),
    Math.hypot(R.x, R.y),
  ];
  const maxMag = Math.max(1, ...magnitudes);


  const scale = 90 / maxMag;


  return (
    <svg
      width="300"
      height="300"
      className="border rounded-lg bg-white shadow mx-auto"
    >
      <g transform="translate(150,150)">
        {/* Axes */}
```

```
        <line x1={-140} y1={0} x2={140} y2={0} stroke="gray"
strokeWidth="1" />
        <line x1={0} y1={-140} x2={0} y2={140} stroke="gray"
strokeWidth="1" />


        {/* Draw each force */}
        {vectors.map((v, i) => {
          const x = v.x * scale;
          const y = -v.y * scale;


          return (
            <g key={i}>
              <line
                x1={0}
                y1={0}
                x2={x}
                y2={y}
                stroke="#1848a0"
                strokeWidth="3"
                markerEnd="url(#arrowF)"
              />
              <text
                x={x * 0.55}
                y={y * 0.55}
                fontSize="14"
                fill="#1848a0"
                fontWeight="bold"
              >
                F{i + 1}
              </text>
            </g>
          );
        })}


        {/* Draw resultant */}
        <line
          x1={0}
          y1={0}
          x2={R.x * scale}
          y2={-R.y * scale}
          stroke="#009900"
          strokeWidth="4"
          markerEnd="url(#arrowR)"
        />
        <text
          x={(R.x * scale) * 0.55}
          y={(-R.y * scale) * 0.55}
          fontSize="16"
```

```
          fill="#009900"
          fontWeight="bold"
        >
          R
        </text>


        {/* Arrow definitions */}
        <defs>
          <marker
            id="arrowF"
            markerWidth="10"
            markerHeight="10"
            refX="5"
            refY="3"
            orient="auto"
          >
            <polygon points="0 0, 6 3, 0 6" fill="#1848a0" />
          </marker>


          <marker
            id="arrowR"
            markerWidth="12"
            markerHeight="12"
            refX="6"
            refY="3"
            orient="auto"
          >
            <polygon points="0 0, 7 3, 0 6" fill="#009900" />
          </marker>
        </defs>
      </g>
    </svg>
  );
}


/*SVG FBD Component (draggable + resultant)*/
function FBD({ forces, setForces }: { forces: ForceInput[];
setForces: (f: ForceInput[]) => void }) {
  const svgRef = useRef<SVGSVGElement | null>(null);
  const [dragIndex, setDragIndex] = useState<number |
null>(null);


  // Convert forces to vectors (math coords; y positive up)
  const vectors = forces
    .map((f) => {
      const m = parseFloat(f.magnitude);
      const a = parseFloat(f.angle);
```

```
      if (isNaN(m) || isNaN(a)) return null;
      const rad = (a * Math.PI) / 180;
      return {
        x: m * Math.cos(rad),
        y: m * Math.sin(rad),
      };
    })
    .filter(Boolean) as { x: number; y: number }[];


  // Determine scale so arrows fit nicely
  const maxMag = Math.max(1, ...vectors.map((v) =>
Math.hypot(v.x, v.y)));
  const scale = 80 / maxMag; // dynamic scale


  const screenPointToSvg = (clientX: number, clientY: number) =>
{
    const svg = svgRef.current;
    if (!svg) return null;
    const pt = svg.createSVGPoint();
    pt.x = clientX;
    pt.y = clientY;
    const ctm = svg.getScreenCTM();
    if (!ctm) return null;
    return pt.matrixTransform(ctm.inverse());
  };


  const handleMouseMove = (e: React.MouseEvent) => {
    if (dragIndex === null) return;


    const svg = svgRef.current;
    if (!svg) return;


    const pt = svg.createSVGPoint();
    pt.x = e.clientX;
    pt.y = e.clientY;


    // convert cursor to SVG coordinates
    const cursor =
pt.matrixTransform(svg.getScreenCTM()?.inverse());
    const x = cursor.x - 150;
    const y = cursor.y - 150;
```

```
    const newAngle = (Math.atan2(-y, x) * 180) / Math.PI;


    const newForces = [...forces];


    newForces[dragIndex] = {
      ...newForces[dragIndex],
      angle: newAngle.toFixed(3),  // Only angle changes
    };


    setForces(newForces);
  };




  const stopDrag = () => setDragIndex(null);


  // compute resultant in math coords
  const sum = vectors.reduce((acc, v) => ({ x: acc.x + v.x, y:
acc.y + v.y }), { x: 0, y: 0 });
  const Rx = sum.x * scale;
  const Ry = -sum.y * scale; // svg y inverted


  return (
    <svg
      ref={svgRef}
      width="300"
      height="300"
      className="border rounded-lg bg-white shadow"
      style={{ background: "white" }}
      onMouseMove={handleMouseMove}
      onMouseUp={stopDrag}
      onMouseLeave={stopDrag}
    >
      <g transform="translate(150,150)">
        {/* Axes */}
        <line x1={-140} y1={0} x2={140} y2={0} stroke="gray"
strokeWidth="1" />
        <line x1={0} y1={-140} x2={0} y2={140} stroke="gray"
strokeWidth="1" />


        {/* Force vectors */}
        {vectors.map((v, i) => {
          const x = v.x * scale;
          const y = -v.y * scale; // invert for svg
```

```
          return (
            <g key={i}>
              <line
                x1={0}
                y1={0}
                x2={x}
                y2={y}
                stroke="#1848a0"
                strokeWidth="3"
                markerEnd="url(#arrow)"
                className="cursor-pointer"
                onMouseDown={() => setDragIndex(i)}
              />
              <text x={x * 0.55} y={y * 0.55} fontSize="14"
fill="black">
                F{i + 1}
              </text>
            </g>
          );
        })}


        {/* Arrow definitions */}
        <defs>
          <marker id="arrow" markerWidth="10" markerHeight="10"
refX="5" refY="3" orient="auto">
            <polygon points="0 0, 6 3, 0 6" fill="#1848a0" />
          </marker>


          <marker id="arrowR" markerWidth="12" markerHeight="12"
refX="6" refY="3" orient="auto">
            <polygon points="0 0, 7 3, 0 6" fill="#009900" />
          </marker>
        </defs>
      </g>
    </svg>


  );
}


/* ==================== MAIN COMPONENT ==================== */
export default function Solver2D() {
  const [forces, setForces] = useState<ForceInput[]>([{
magnitude: "", angle: "" }]);


  const [result, setResult] = useState<ForceResult | null>(null);
```

```
  const handleInputChange = (index: number, field: "magnitude" |
"angle", value: string) => {
    const newForces = [...forces];
    newForces[index][field] = value;
    setForces(newForces);
  };


  const calculateResultant = () => {
    const system = new ForceSystem2D();


    forces.forEach((f) => {
      const mag = parseFloat(f.magnitude);
      const ang = parseFloat(f.angle);
      if (!isNaN(mag) && !isNaN(ang)) system.addForce(mag, ang);
    });


    setResult(system.stepByStepSolution());
  };


  return (
    <div className="flex flex-col min-h-screen bg-gray-50
text-gray-900 text-[18px]">
      <Header />


      <main className="flex-grow flex flex-col items-center px-4
py-10">
        <h1 className="text-[32px] font-bold mb-6">2D Resultant
Force Calculator</h1>


        {/* FBD Live Preview */}
        <div className="mb-8">
          <h2 className="text-[20px] font-semibold text-center
mb-2">Real-Time Free Body Diagram</h2>
          <FBD forces={forces} setForces={setForces} />
        </div>


        {/* Inputs */}
        <div className="w-full max-w-xl bg-white rounded-2xl
shadow p-6 space-y-6">
          <h2 className="text-[20px] font-semibold">Force
setup</h2>
```

```
          <div className="grid grid-cols-2 gap-4">
            {forces.map((f, i) => (
              <div key={i} className="col-span-2 flex gap-4
items-end">
                <div className="flex-1">
                  <label className="block font-medium
text-[18px]">
                    Force {i + 1} (kN)
                  </label>
                  <input
                    type="number"
                    value={f.magnitude}
                    onChange={(e) =>
                      handleInputChange(i, "magnitude",
e.target.value)
                    }
                    placeholder="Magnitude (kN)"
                    className="w-full mt-1 rounded-lg
border-gray-300 text-[18px] p-2"
                  />
                </div>
                <div className="flex-1">
                  <label className="block font-medium
text-[18px]">
                    Angle {i + 1} (°)
                  </label>
                  <input
                    type="number"
                    value={f.angle}
                    onChange={(e) =>
                      handleInputChange(i, "angle",
e.target.value)
                    }
                    placeholder="Angle (deg)"
                    className="w-full mt-1 rounded-lg
border-gray-300 text-[18px] p-2"
                  />
                </div>
                {forces.length > 1 && (
                  <button
                    onClick={() =>
                      setForces(forces.filter((_, idx) => idx !==
i))
                    }
                    className="px-3 py-1 bg-red-500 text-white
rounded-lg hover:bg-red-600 text-[18px]"
                  >
                    -
                  </button>
                )}
              </div>
```

```
          ))}
        </div>
        <button onClick={() => setForces([...forces, {
magnitude: "", angle: "" }])} className="w-full bg-[#008409]
text-white py-3 rounded-lg hover:bg-[#15711b] transition
text-[18px]">
          + Add Force
        </button>


        <button onClick={calculateResultant} className="w-full
bg-[#1848a0] text-white py-3 rounded-lg hover:bg-[#163d8a]
transition text-[18px]">
          Calculate
        </button>
      </div>


      {/* Output */}
      {result && (
        <div className="w-full max-w-xl mt-6 bg-white
rounded-2xl shadow p-6 space-y-4">
          <h2 className="text-[20px] font-semibold">Resultant
Force (kN)</h2>
          <div>
            <label className="block font-medium
text-[18px]">Horizontal component (Fx)</label>
            <input type="text"
value={`${result.sumFx.toFixed(3)} kN`} readOnly
className="w-full mt-1 rounded-lg border-gray-300 text-[18px]
p-2" />
          </div>


          <div>
            <label className="block font-medium
text-[18px]">Vertical component (Fy)</label>
            <input type="text"
value={`${result.sumFy.toFixed(3)} kN`} readOnly
className="w-full mt-1 rounded-lg border-gray-300 text-[18px]
p-2" />
          </div>


          <div>
            <label className="block font-medium
text-[18px]">Magnitude of resultant force (R)</label>
            <input type="text" value={`${result.R.toFixed(3)}
kN`} readOnly className="w-full mt-1 rounded-lg border-gray-300
text-[18px] p-2" />
          </div>
```

```
            <div>
              <label className="block font-medium
text-[18px]">Direction of resultant force (θ)</label>
              <input type="text"
value={`${result.theta.toFixed(2)}°`} readOnly className="w-full
mt-1 rounded-lg border-gray-300 text-[18px] p-2" />
            </div>
          </div>
        )}


        {/* Step-by-Step Solution */}
        {result && (
          <div className="w-full max-w-xl mt-6 bg-white
rounded-2xl shadow p-6">
            <h2 className="text-[20px] font-semibold
mb-2">Step-by-Step Solution</h2>




            <div className="space-y-4">
              {result.steps.map((line, i) =>
                line.startsWith("Step") ? (
                  <p key={i} className="font-medium text-[18px]">
                    {line}
                  </p>
                ) : (
                  <div key={i} className="text-[18px]">
                    <BlockMath>{line}</BlockMath>
                  </div>
                )
              )}
            </div>


            {/* Step 4 */}
            <div className="mt-8">
              <p className="font-medium text-[18px] mb-2">
                Step 4: Final Free Body Diagram
              </p>
              <ResultantFBD forces={forces} result={result} />
            </div>
          </div>
        )}
      </main>


      <Footer />
```

```
    </div>
  );
}

"use client";

import { useRef, useState } from "react";
import Header from "../../components/Header";
import Footer from "../../components/Footer";
import "katex/dist/katex.min.css";
import { BlockMath } from "react-katex";

/* ===================== Force System Logic =====================
*/
class ForceSystem2D {
  vectors: { fx: number; fy: number; magnitude: number; angleDeg:
number }[];

  constructor() {
    this.vectors = [];
  }

  addForce(magnitude: number, angleDeg: number) {
    const angleRad = (angleDeg * Math.PI) / 180;
    const fx = magnitude * Math.cos(angleRad);
    const fy = magnitude * Math.sin(angleRad);
    this.vectors.push({ fx, fy, magnitude, angleDeg });
  }

  stepByStepSolution() {
    const steps: string[] = [];
    steps.push("Step 1: Resolve each force into components:");

    let sumFx = 0;
    let sumFy = 0;

    this.vectors.forEach((v, i) => {
      steps.push(
        `\\text{Force ${i + 1}: }
|F|=${v.magnitude}\\,\\text{kN},\\; \\theta=${v.angleDeg}^\\circ`
      );
```

```
      steps.push(`
        \\begin{align*}
        F_{x${i + 1}} &=
${v.magnitude}\\cos(${v.angleDeg}^\\circ) \\\\
                      &= ${v.fx.toFixed(3)}\\,\\text{kN} \\\\
        F_{y${i + 1}} &=
${v.magnitude}\\sin(${v.angleDeg}^\\circ) \\\\
                      &= ${v.fy.toFixed(3)}\\,\\text{kN}
        \\end{align*}
      `);


      sumFx += v.fx;
      sumFy += v.fy;
    });


    steps.push("Step 2: Sum of components:");
    steps.push(`
      \\begin{align*}
      \\Sigma F_x &= ${sumFx.toFixed(3)}\\,\\text{kN} \\\\
      \\Sigma F_y &= ${sumFy.toFixed(3)}\\,\\text{kN}
      \\end{align*}
    `);


    const R = Math.hypot(sumFx, sumFy);
    const theta = (Math.atan2(sumFy, sumFx) * 180) / Math.PI;


    const arrow = theta >= 0 ? "↺" : "↻";


    steps.push("Step 3: Resultant force:");
    steps.push(`
      \\begin{align*}
      R &= \\sqrt{(\\Sigma F_x)^2 + (\\Sigma F_y)^2} \\\\
        &= ${R.toFixed(3)}\\,\\text{kN} \\\\
      \\theta &= \\tan^{-1}\\left(\\tfrac{\\Sigma F_y}{\\Sigma
F_x}\\right) \\\\
              &= ${theta.toFixed(2)}^\\circ
${arrow}\\,\\text{from +x axis}
      \\end{align*}
    `);


    return { steps, sumFx, sumFy, R, theta };
  }
}
```

```
type ForceInput = {
  magnitude: string;
  angle: string;
};


type ForceResult = {
  steps: string[];
  sumFx: number;
  sumFy: number;
  R: number;
  theta: number;
};


/*FULL FBD FOR STEP 4 (all forces + resultant)*/
function ResultantFBD({
  forces,
  result,
}: {
  forces: ForceInput[];
  result: ForceResult;
}) {
  const vectors = forces
    .map((f) => {
      const m = parseFloat(f.magnitude);
      const a = parseFloat(f.angle);
      if (isNaN(m) || isNaN(a)) return null;
      const rad = (a * Math.PI) / 180;
      return { x: m * Math.cos(rad), y: m * Math.sin(rad) };
    })
    .filter(Boolean) as { x: number; y: number }[];


  const R = { x: result.sumFx, y: result.sumFy };


  const magnitudes = [
    ...vectors.map((v) => Math.hypot(v.x, v.y)),
    Math.hypot(R.x, R.y),
  ];
  const maxMag = Math.max(1, ...magnitudes);


  const scale = 90 / maxMag;


  return (
    <svg
      width="300"
      height="300"
```

```
      className="border rounded-lg bg-white shadow mx-auto"
    >
      <g transform="translate(150,150)">
        {/* Axes */}
        <line x1={-140} y1={0} x2={140} y2={0} stroke="gray"
strokeWidth="1" />
        <line x1={0} y1={-140} x2={0} y2={140} stroke="gray"
strokeWidth="1" />


        {/* Draw each force */}
        {vectors.map((v, i) => {
          const x = v.x * scale;
          const y = -v.y * scale;


          return (
            <g key={i}>
              <line
                x1={0}
                y1={0}
                x2={x}
                y2={y}
                stroke="#1848a0"
                strokeWidth="3"
                markerEnd="url(#arrowF)"
              />
              <text
                x={x * 0.55}
                y={y * 0.55}
                fontSize="14"
                fill="#1848a0"
                fontWeight="bold"
              >
                F{i + 1}
              </text>
            </g>
          );
        })}


        {/* Draw resultant */}
        <line
          x1={0}
          y1={0}
          x2={R.x * scale}
          y2={-R.y * scale}
          stroke="#009900"
          strokeWidth="4"
          markerEnd="url(#arrowR)"
        />
```

```
        <text
          x={(R.x * scale) * 0.55}
          y={(-R.y * scale) * 0.55}
          fontSize="16"
          fill="#009900"
          fontWeight="bold"
        >
          R
        </text>


        {/* Arrow definitions */}
        <defs>
          <marker
            id="arrowF"
            markerWidth="10"
            markerHeight="10"
            refX="5"
            refY="3"
            orient="auto"
          >
            <polygon points="0 0, 6 3, 0 6" fill="#1848a0" />
          </marker>


          <marker
            id="arrowR"
            markerWidth="12"
            markerHeight="12"
            refX="6"
            refY="3"
            orient="auto"
          >
            <polygon points="0 0, 7 3, 0 6" fill="#009900" />
          </marker>
        </defs>
      </g>
    </svg>
  );
}


/*SVG FBD Component (draggable + resultant)*/
function FBD({ forces, setForces }: { forces: ForceInput[];
setForces: (f: ForceInput[]) => void }) {
  const svgRef = useRef<SVGSVGElement | null>(null);
  const [dragIndex, setDragIndex] = useState<number |
null>(null);


  // Convert forces to vectors (math coords; y positive up)
```

```
  const vectors = forces
    .map((f) => {
      const m = parseFloat(f.magnitude);
      const a = parseFloat(f.angle);
      if (isNaN(m) || isNaN(a)) return null;
      const rad = (a * Math.PI) / 180;
      return {
        x: m * Math.cos(rad),
        y: m * Math.sin(rad),
      };
    })
    .filter(Boolean) as { x: number; y: number }[];


  // Determine scale so arrows fit nicely
  const maxMag = Math.max(1, ...vectors.map((v) =>
Math.hypot(v.x, v.y)));
  const scale = 80 / maxMag; // dynamic scale


  const screenPointToSvg = (clientX: number, clientY: number) =>
{
    const svg = svgRef.current;
    if (!svg) return null;
    const pt = svg.createSVGPoint();
    pt.x = clientX;
    pt.y = clientY;
    const ctm = svg.getScreenCTM();
    if (!ctm) return null;
    return pt.matrixTransform(ctm.inverse());
  };


  const handleMouseMove = (e: React.MouseEvent) => {
    if (dragIndex === null) return;


    const svg = svgRef.current;
    if (!svg) return;


    const pt = svg.createSVGPoint();
    pt.x = e.clientX;
    pt.y = e.clientY;


    // convert cursor to SVG coordinates
    const cursor =
pt.matrixTransform(svg.getScreenCTM()?.inverse());
    const x = cursor.x - 150;
    const y = cursor.y - 150;
```

```
    const newAngle = (Math.atan2(-y, x) * 180) / Math.PI;


    const newForces = [...forces];


    newForces[dragIndex] = {
      ...newForces[dragIndex],
      angle: newAngle.toFixed(3),  // Only angle changes
    };


    setForces(newForces);
  };




  const stopDrag = () => setDragIndex(null);


  // compute resultant in math coords
  const sum = vectors.reduce((acc, v) => ({ x: acc.x + v.x, y:
acc.y + v.y }), { x: 0, y: 0 });
  const Rx = sum.x * scale;
  const Ry = -sum.y * scale; // svg y inverted


  return (
    <svg
      ref={svgRef}
      width="300"
      height="300"
      className="border rounded-lg bg-white shadow"
      style={{ background: "white" }}
      onMouseMove={handleMouseMove}
      onMouseUp={stopDrag}
      onMouseLeave={stopDrag}
    >
      <g transform="translate(150,150)">
        {/* Axes */}
        <line x1={-140} y1={0} x2={140} y2={0} stroke="gray"
strokeWidth="1" />
        <line x1={0} y1={-140} x2={0} y2={140} stroke="gray"
strokeWidth="1" />
```

```
        {/* Force vectors */}
        {vectors.map((v, i) => {
          const x = v.x * scale;
          const y = -v.y * scale; // invert for svg
          return (
            <g key={i}>
              <line
                x1={0}
                y1={0}
                x2={x}
                y2={y}
                stroke="#1848a0"
                strokeWidth="3"
                markerEnd="url(#arrow)"
                className="cursor-pointer"
                onMouseDown={() => setDragIndex(i)}
              />
              <text x={x * 0.55} y={y * 0.55} fontSize="14"
fill="black">
                F{i + 1}
              </text>
            </g>
          );
        })}


        {/* Arrow definitions */}
        <defs>
          <marker id="arrow" markerWidth="10" markerHeight="10"
refX="5" refY="3" orient="auto">
            <polygon points="0 0, 6 3, 0 6" fill="#1848a0" />
          </marker>


          <marker id="arrowR" markerWidth="12" markerHeight="12"
refX="6" refY="3" orient="auto">
            <polygon points="0 0, 7 3, 0 6" fill="#009900" />
          </marker>
        </defs>
      </g>
    </svg>


  );
}


/* MAIN COMPONENT */
export default function Solver2D() {
  const [forces, setForces] = useState<ForceInput[]>([{
magnitude: "", angle: "" }]);
```

```
  const [result, setResult] = useState<ForceResult | null>(null);


  const handleInputChange = (index: number, field: "magnitude" |
"angle", value: string) => {
    const newForces = [...forces];
    newForces[index][field] = value;
    setForces(newForces);
  };


  const calculateResultant = () => {
    const system = new ForceSystem2D();


    forces.forEach((f) => {
      const mag = parseFloat(f.magnitude);
      const ang = parseFloat(f.angle);
      if (!isNaN(mag) && !isNaN(ang)) system.addForce(mag, ang);
    });


    setResult(system.stepByStepSolution());
  };


  return (
    <div className="flex flex-col min-h-screen bg-gray-50
text-gray-900 text-[18px]">
      <Header />


      <main className="flex-grow flex flex-col items-center px-4
py-10">
        <h1 className="text-[32px] font-bold mb-6">2D Resultant
Force Calculator</h1>


        {/* FBD Live Preview */}
        <div className="mb-8">
          <h2 className="text-[20px] font-semibold text-center
mb-2">Real-Time Free Body Diagram</h2>
          <FBD forces={forces} setForces={setForces} />
        </div>


        {/* Inputs */}
        <div className="w-full max-w-xl bg-white rounded-2xl
shadow p-6 space-y-6">
```

```
          <h2 className="text-[20px] font-semibold">Force
setup</h2>


          <div className="grid grid-cols-2 gap-4">
            {forces.map((f, i) => (
              <div key={i} className="col-span-2 flex gap-4
items-end">
                <div className="flex-1">
                  <label className="block font-medium
text-[18px]">
                    Force {i + 1} (kN)
                  </label>
                  <input
                    type="number"
                    value={f.magnitude}
                    onChange={(e) =>
                      handleInputChange(i, "magnitude",
e.target.value)
                    }
                    placeholder="Magnitude (kN)"
                    className="w-full mt-1 rounded-lg
border-gray-300 text-[18px] p-2"
                  />
                </div>
                <div className="flex-1">
                  <label className="block font-medium
text-[18px]">
                    Angle {i + 1} (°)
                  </label>
                  <input
                    type="number"
                    value={f.angle}
                    onChange={(e) =>
                      handleInputChange(i, "angle",
e.target.value)
                    }
                    placeholder="Angle (deg)"
                    className="w-full mt-1 rounded-lg
border-gray-300 text-[18px] p-2"
                  />
                </div>
                {forces.length > 1 && (
                  <button
                    onClick={() =>
                      setForces(forces.filter((_, idx) => idx !==
i))
                    }
                    className="px-3 py-1 bg-red-500 text-white
rounded-lg hover:bg-red-600 text-[18px]"
                  >
```

```
                  -
                </button>
              )}
            </div>
          ))}
        </div>
        <button onClick={() => setForces([...forces, {
magnitude: "", angle: "" }])} className="w-full bg-[#008409]
text-white py-3 rounded-lg hover:bg-[#15711b] transition
text-[18px]">
          + Add Force
        </button>


        <button onClick={calculateResultant} className="w-full
bg-[#1848a0] text-white py-3 rounded-lg hover:bg-[#163d8a]
transition text-[18px]">
          Calculate
        </button>
      </div>


      {/* Output */}
      {result && (
        <div className="w-full max-w-xl mt-6 bg-white
rounded-2xl shadow p-6 space-y-4">
          <h2 className="text-[20px] font-semibold">Resultant
Force (kN)</h2>
          <div>
            <label className="block font-medium
text-[18px]">Horizontal component (Fx)</label>
            <input type="text"
value={`${result.sumFx.toFixed(3)} kN`} readOnly
className="w-full mt-1 rounded-lg border-gray-300 text-[18px]
p-2" />
          </div>


          <div>
            <label className="block font-medium
text-[18px]">Vertical component (Fy)</label>
            <input type="text"
value={`${result.sumFy.toFixed(3)} kN`} readOnly
className="w-full mt-1 rounded-lg border-gray-300 text-[18px]
p-2" />
          </div>


          <div>
            <label className="block font-medium
text-[18px]">Magnitude of resultant force (R)</label>
```

```
              <input type="text" value={`${result.R.toFixed(3)} kN`} readOnly className="w-full mt-1 rounded-lg border-gray-300 text-[18px] p-2" />
            </div>


            <div>
              <label className="block font-medium text-[18px]">Direction of resultant force (θ)</label>
              <input type="text" value={`${result.theta.toFixed(2)}°`} readOnly className="w-full mt-1 rounded-lg border-gray-300 text-[18px] p-2" />
            </div>
          </div>
        )}


        {/* Step-by-Step Solution */}
        {result && (
          <div className="w-full max-w-xl mt-6 bg-white rounded-2xl shadow p-6">
            <h2 className="text-[20px] font-semibold mb-2">Step-by-Step Solution</h2>




            <div className="space-y-4">
              {result.steps.map((line, i) =>
                line.startsWith("Step") ? (
                  <p key={i} className="font-medium text-[18px]">
                    {line}
                  </p>
                ) : (
                  <div key={i} className="text-[18px]">
                    <BlockMath>{line}</BlockMath>
                  </div>
                )
              )}
            </div>


            {/* Step 4 */}
            <div className="mt-8">
              <p className="font-medium text-[18px] mb-2">
                Step 4: Final Free Body Diagram
              </p>
              <ResultantFBD forces={forces} result={result} />
            </div>
          </div>
        )}
```

```
      </main>


      <Footer />
    </div>
  );
}
```

Footer

```
"use client";


import Link from "next/link";


type NavLink = { label: string; href: string };


interface FooterProps {
  links?: NavLink[];
}


export default function Footer({
  links = [
    { label: "About", href: "/about" },
    { label: "References", href: "/reference" },
    { label: "Contact", href: "/contact" },
    { label: "Developer", href: "/developers" },
  ],
}: FooterProps) {
  return (
    <footer className="bg-white border-t mt-auto">
      <div className="max-w-7xl mx-auto px-6 py-4 text-center
text-gray-700 text-[18px]">
        {/* Desktop: horizontal links with | separator */}
        <div className="hidden sm:flex justify-center flex-wrap
gap-4">
          {links.map((link, idx) => (
            <span key={link.href} className="flex items-center
text-[18px]">

              <Link href={link.href}
className="hover:text-blue-600">
                {link.label}
              </Link>
              {idx < links.length - 1 && (
                <span className="mx-2 text-gray-400">|</span>
              )}
            </span>
          ))}
        </div>


        {/* Mobile: stacked links */}
        <div className="flex flex-col sm:hidden gap-2">
          {links.map((link) => (
            <Link
```

```
              key={link.href}
              href={link.href}
              className="hover:text-blue-600 text-[18px]"
            >
              {link.label}
            </Link>
          ))}
        </div>
      </div>
    </footer>
  );
}
```

Header

```
"use client";


import Link from "next/link";


export default function Header() {
  return (
    <header className="bg-white shadow">
      <div className="max-w-7xl mx-auto px-6 py-4 flex flex-col
sm:flex-row sm:items-center sm:justify-between gap-4">


        {/* Logo + Title */}
        <Link
          href="/"
          className="flex items-center gap-3 justify-center
sm:justify-start hover:text-[#1848a0] transition"
        >
          <div className="w-10 h-10 border-2 border-black
rounded-full" />
          <span className="font-bold text-[30px]
text-black">Statics Calculator</span>
        </Link>


        {/* Desktop Navigation */}
        <nav className="hidden sm:flex items-center space-x-6
text-gray-700 relative text-[18px]">
          <Link href="/"
className="hover:text-[#1848a0]">Home</Link>
          <span>|</span>

          {/* Topics Dropdown */}
          <div className="group relative">
            <button className="hover:text-[#1848a0]">Topics
▾</button>
            <div

              className="absolute left-1/2 -translate-x-1/2 mt-2
w-56 bg-white border rounded-lg shadow-lg
                opacity-0 group-hover:opacity-100 invisible
group-hover:visible transition text-[18px]"
            >
              <div className="flex flex-col p-2 text-gray-700">
                <Link href="/Introduction"
className="hover:text-[#1848a0] p-2">
                  Chapter 1: Introduction to Statics
                </Link>
```

```
                <Link href="/2D-solver"
className="hover:text-[#1848a0] p-2">
                  Chapter 2: Force Systems
                </Link>
                <Link href="/Equilibrium"
className="hover:text-[#1848a0] p-2">
                  Chapter 3: Equilibrium
                </Link>
                <Link href="/Structures"
className="hover:text-[#1848a0] p-2">
                  Chapter 4: Structures
                </Link>
                <Link href="/Distributed-Loads"
className="hover:text-[#1848a0] p-2">
                  Chapter 5: Distributed Loads
                </Link>
              </div>
            </div>
          </div>


          <span>|</span>
          <Link href="/about"
className="hover:text-[#1848a0]">About</Link>
        </nav>


        {/* Mobile Navigation */}
        <nav className="flex flex-col sm:hidden items-center
gap-2 text-gray-700 text-[18px]">
          <Link href="/"
className="hover:text-[#1848a0]">Home</Link>


          <details className="w-full">
            <summary className="cursor-pointer text-center
hover:text-[#1848a0]">Topics</summary>
            <div className="flex flex-col mt-2 gap-2">
              <Link href="/Introduction"
className="hover:text-[#1848a0] p-2">
                Chapter 1: Introduction to Statics
              </Link>
              <Link href="/2D-solver"
className="hover:text-[#1848a0] p-2">
                Chapter 2: Force Systems
              </Link>
              <Link href="/Equilibrium"
className="hover:text-[#1848a0] p-2">
                Chapter 3: Equilibrium
              </Link>
```

```
              <Link href="/Structures"
className="hover:text-[#1848a0] p-2">
                Chapter 4: Structures
              </Link>
              <Link href="/Distributed-Loads"
className="hover:text-[#1848a0] p-2">
                Chapter 5: Distributed Loads
              </Link>
            </div>
          </details>


          <Link href="/about"
className="hover:text-[#1848a0]">About</Link>
        </nav>
      </div>
    </header>
  );
}
```

About page

```
import Header from "<Ian>/components/Header";
import Footer from "<Ian>/components/Footer";


export default function AboutPage() {
  return (
    <div className="min-h-screen flex flex-col bg-gray-50">
      <Header />


      {/* Main Content */}
      <main className="flex flex-1 items-center justify-center
px-6 py-12">
        <div className="max-w-3xl text-center">
          <h1 className="text-2xl font-semibold text-gray-800
mb-6">
            About{" "}
            <span className="text-[#1848a0]">StatiCalcs</span>
          </h1>


          <p className="text-[18px] text-gray-700 leading-relaxed
mb-6">
            <span className="font-bold">
              Stati<span className="text-[#1848a0]">Calcs</span>
            </span>{" "}
            is an interactive web-based learning tool created to
support
            engineering students in their study of Statics of
Rigid Bodies. It
            combines essential concepts with integrated
calculators to help
            users practice problem-solving more effectively.
          </p>

         <p className="text-[18px] text-gray-700
leading-relaxed">
  Designed specifically for engineering students of MSU-Gensan,{"
"}
  <span className="font-bold">
    Stati<span className="text-[#1848a0]">Calcs</span>
  </span>{" "}
  serves as a supplementary academic tool that enhances classroom
learning,
  encourages independent study, and fosters a deeper
understanding of
  statics principles.
</p>
```

```
        </div>
      </main>


      <Footer />
    </div>
  );
}
```

Page Contact

```
import Header from "<Ian>/components/Header";
import Footer from "<Ian>/components/Footer";


export default function ContactPage() {
  return (
    <div className="min-h-screen flex flex-col bg-gray-50">
      <Header />


      {/* Main Content */}
      <main className="flex flex-1 items-center justify-center
px-6 py-12">
        <div className="max-w-3xl text-center">
          <h1 className="text-2xl font-semibold text-gray-800
mb-4">
            Contact
          </h1>
          <p className="text-[18px] text-gray-700 mb-8">
            For feedback or inquiries, please reach out through
the following:
          </p>

          <div className="space-y-8 text-left">
            {/* First Contact */}
            <div>
              <h2 className="font-semibold text-[18px]
text-gray-800">
                Ian Carl P. Cona
              </h2>
              <p className="text-[18px] text-gray-700">
                Email: iancarl.cona@msugensan.edu.ph
              </p>
              <p className="text-[18px] text-gray-700">
                Mindanao State University – General Santos
              </p>

              <p className="text-[18px] text-gray-700">
                Fatima, General Santos City, Philippines
              </p>
            </div>


            {/* Second Contact */}
            <div>
              <h2 className="font-semibold text-[18px]
text-gray-800">
                Sophia Daphne C. Faelnar
```

```
              </h2>
              <p className="text-[18px] text-gray-700">
                Email: sophiadaphne.faelnar@msugensan.edu.ph
              </p>
              <p className="text-[18px] text-gray-700">
                Mindanao State University – General Santos
              </p>
              <p className="text-[18px] text-gray-700">
                Fatima, General Santos City, Philippines
              </p>
            </div>
          </div>
        </div>
      </main>


      <Footer />
    </div>
  );
}
```