

# Improving Cryptography Education through Adaptive Web Interfaces: Usability, Accessibility, and Interactive Learning

Saurav Ghosh

Southeast Missouri State University  
Cape Girardeau, Missouri, USA

Suhair Amer

Southeast Missouri State University  
Cape Girardeau, Missouri, USA

## ABSTRACT

Current educational tools for teaching basic encryption methods, such as the Caesar, Monoalphabetic, and Vigenère ciphers, typically rely on static interfaces, which do not accommodate varied user needs or accessibility requirements. This paper describes a class project development of an adaptive web-based learning application specifically tailored to simplify foundational encryption concepts for beginners, particularly undergraduate and entry-level learners with limited prior cryptography experience. The interface dynamically adjusts instructional support based on real-time user interactions, offering adaptive hints, immediate validation feedback, and interactive quizzes designed to enhance engagement and comprehension. For proof of concept, usability evaluations were conducted with 10 undergraduate participants who had minimal prior cryptography knowledge. Qualitative feedback and quantitative assessments demonstrated high user satisfaction regarding navigability, visual appeal, clarity of instructions, and perceived effectiveness of adaptive features. However, participants also highlighted areas for improvement, including clearer feedback for input errors and more explicit guidance on adaptive functionality. Although the current implementation is tailored specifically for introductory cryptography concepts, our adaptive approach can be extended and scaled to more complex cryptographic algorithms or other STEM subjects. This research contributes a practical model illustrating how adaptive interfaces can effectively enhance learning experiences through dynamic responsiveness to user needs.

## Keywords

Adaptive interfaces, cryptography education, accessibility, usability testing, interactive learning, user-centered design

## 1. INTRODUCTION

Adaptive user interfaces are becoming very important in making web-based learning more effective, enjoyable, and user-friendly. These interfaces smartly change their content and layout depending on how the user interacts with them, creating personalized learning experiences. In this research, we explore how adaptive design principles can be effectively applied to teach basic cryptographic concepts through an interactive online platform. Specifically, our focus is on three foundational encryption techniques: the Caesar Cipher, Monoalphabetic Cipher, and Vigenère Cipher.

Previous studies have shown clearly that adaptive user interfaces significantly improve usability by customizing content according to what users prefer and how they behave [4]. Such adaptability is especially helpful in meeting the

needs of diverse users who may be using different devices or working in different environments [5]. One major challenge in building these adaptive systems is accurately understanding user interactions. Earlier research has explored various effective methods for studying user behavior, creating a strong foundation for developing adaptive interfaces [8]. Additionally, simple rule-based approaches have been effectively used to provide responsive interactions, making applications more engaging [7].

The introduction of machine learning techniques has further improved adaptive interfaces, enabling these systems to continuously learn and adjust according to user interactions [13, 14]. Personalized recommendations powered by adaptive mechanisms have greatly increased user satisfaction and engagement [11]. Such adaptive methods have been beneficial in specialized fields as well; for instance, custom support tools have significantly improved user performance in healthcare tasks [12]. Similarly, adaptive learning systems have successfully enhanced accessibility and improved learning experiences, especially for users with different abilities and learning styles [3].

This paper explores how adaptive interfaces can improve learning in cryptography. It investigates whether immediate visual feedback and real-time validation can simplify complex cryptographic concepts, making it easier for beginners to understand. Further, it examines if adaptive tutorials and interactive hints can reduce user frustration, help retain information better and make cryptography more approachable for everyone. This research also considers how accessibility-focused design choices might attract a wider variety of users, accommodating different learning needs. Additionally, it discusses how an adaptive approach could be scaled up using advanced techniques like machine learning to handle more complex cryptographic methods. To create effective adaptive user interfaces, several structured frameworks have already been proposed. For instance, USIXML provides a model-based approach that helps design flexible interfaces, easily adapting to changing user needs [17, 18]. Additionally, user-modeling frameworks like the GUMO ontology closely track user preferences and behaviors, helping deliver highly personalized experiences [9]. Human-centered design principles, emphasizing ease of use and intuitive interaction, remain central to successful adaptive interface development [10]. Furthermore, mixed initiative methods allow users to have greater personal control over adaptations, enhancing engagement and satisfaction [19].

Evaluating user experience is essential in adaptive interface design, as it helps identify both strengths and areas for improvement. Tools such as questionnaires and direct user

testing are valuable for continuously adapting interfaces to match user expectations and requirements [15, 16].

This paper presents an interactive, web-based adaptive interface designed specifically for teaching cryptography. The adaptive system dynamically responds to real-time user interactions, customizing content and providing targeted feedback, creating a more engaging learning experience [6]. It also emphasizes cultural adaptability by including multilingual support and local examples, making the platform relevant for a global audience [2]. Additionally, user modeling techniques from adaptive hypermedia are used to tailor learning according to individual user behavior and preferences, further improving educational effectiveness [1].

This study will demonstrate how adaptive interfaces offer a powerful method to enhance traditional cryptography education. By combining adaptive learning principles, strong accessibility standards, and user-centered design, this research provides practical insights for building effective, inclusive, and enjoyable educational technologies.

## **2. RESEARCH GOALS AND SCOPE**

The main goal of this research is to create an interactive, adaptive, and web-based learning application to make basic encryption concepts easy, accessible, and engaging for learners. This work aims at helping students, beginners in cybersecurity, or anyone interested in encryption methods to understand cryptographic ideas better. By converting theory into practical, hands-on experiences, the application hopes to make learning cryptography simple, interesting, and effective.

One important objective is to allow direct interaction with simple encryption techniques like the Caesar Cipher, Monoalphabetic Cipher, and Vigenère Cipher. Users can enter plaintext messages, choose shift values or substitution keys, and instantly see how these changes affect the results. Interactive tutorials support this learning process, providing step-by-step guidance and historical background to help users better understand the logic behind encryption methods. Such interactive learning can help users actively engage with the content, enhancing their practical knowledge.

Another major goal is maintaining user engagement throughout the learning experience. To achieve this, the web application includes dynamic visual elements, animations, and immediate feedback. Learners can test their understanding through quizzes and receive real-time guidance if they face difficulties. Such features keep users motivated; help build their confidence and provide a clear sense of progress. A smooth and intuitive user interface ensures that learning remains enjoyable and rewarding.

Accessibility for all types of users is also a priority for this research. The platform includes support for screen readers, keyboard navigation, and multilingual content, enabling users from different backgrounds and abilities to learn comfortably. Features like adjustable font sizes, high-contrast themes, and clear error messages ensure the application is user-friendly and accessible.

Performance optimization ensures the application functions smoothly on various devices, including desktops and mobile phones, by minimizing delays during encryption and decryption tasks. Efficient design helps learners navigate easily between different learning resources, such as tutorials and quizzes.

Through these objectives, our research aims to create an educational tool that not only teaches cryptography but also inspires users to actively engage, experiment, and deepen their understanding.

## **2.1 Role of Small-Scale Projects in Cryptography Education**

Completing small-scale projects in courses offers a variety of benefits to students, both in terms of academic development and personal growth. These smaller-scale assignments can have a significant impact on learning and provide key skills that are useful in larger projects and professional environments. Some of the key benefits include:

- Encourages Active Learning.
- Build Time Management Skills.
- Fosters Creativity and Innovation.
- Develops Problem-Solving Skills.
- Improves Focus and Concentration.
- Boosts Confidence.
- Prepares for Larger Projects.
- Encourages Collaboration.
- Promote Continuous Feedback.
- Increase Engagement and Motivation.
- Prepares for the Professional Environment.
- Improve Adaptability.

In summary, small projects in courses provide numerous benefits that help students develop a wide range of skills, from time management and problem-solving to creativity and teamwork. They create opportunities for students to apply what they've learned, receive feedback, and build confidence, ultimately setting them up for success in larger projects and future professional endeavors. These projects enhance learning by promoting active engagement, self-directed learning, and the development of transferable skills.

## **3. METHODOLOGY AND SYSTEM DESIGN**

The methodology used in this research combined adaptive design principles, user-centered methods, and continuous improvement based on feedback. The focus was on creating an interactive web application that enhances learning, usability, and accessibility specifically for cryptography education.

As illustrated in Figure 1, the method follows a continuous improvement cycle. Users start by entering plaintext or ciphertext along with details like shift values or substitution keys. The web application immediately processes these inputs, displaying results and checking for errors. If users face any difficulties, the adaptive interface quickly provides helpful hints or guides users to relevant tutorials.

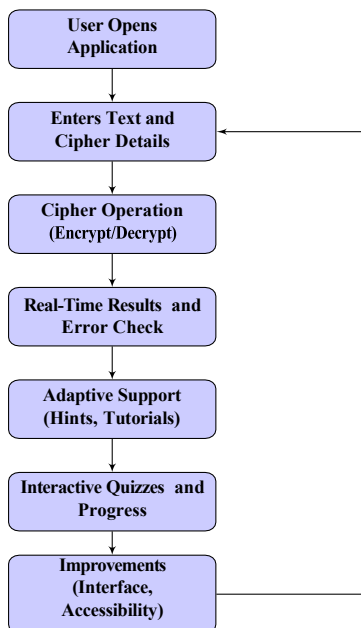
The application regularly assesses users' understanding through interactive quizzes and progress tracking. Data from user interactions helps the system make improvements. These improvements include making the interface simpler, enhancing accessibility, clarifying error messages, and introducing advanced cipher options for experienced learners.

The initial design clearly defined the educational goals based on user needs. Early versions allowed users to directly interact with ciphers, instantly seeing results and understanding how encryption works practically. Tutorials were designed to simplify complex ideas through clear examples and historical context. All input fields were clearly labeled to help beginners easily

follow instructions.

Keeping users engaged was essential. Therefore, interactive animations, real-time feedback, and colorful visual elements were used. Adaptive features monitored user performance and provided extra support when needed, reducing frustration and boosting confidence.

Accessibility was a top priority. The application supported screen readers for visually impaired users and allowed easy navigation through the keyboard for those with motor skill limitations. Multilingual options were available, and users could customize font size, contrast, and animations to suit their individual needs. In this study testing did not cover every possible physical or cultural barrier, so absolute claims about complete accessibility were avoided.



**Fig 1: Adaptive Learning Flow: Users Input Encryption Details, Receive Instant Feedback and Adaptive Hints, and Progress Is Assessed Through Interactive Quizzes.**

### 3.1 Interactive Cryptography learning system

Practical implementation was accomplished using HTML, CSS, and JavaScript. The following code snippets illustrate how core encrypting algorithms were implemented within the interactive application.

Next is the code for *Caesar Cipher Algorithm*. The Caesar Cipher shifts characters based on a user-defined numeric input

```

function encryptCaesar() {
  const input = document.getElementById('caesar-input').value;
  const shift =
    <math>\rightarrow</math> parseInt(document.getElementById('caesar-shift').value) ||
    <math>\rightarrow</math> 0;
  let result = "";
  for (let char of input) {
    if (char.match(/[a-z]/i)) {
      const code = char.charCodeAt(0);
    }
  }
}
  
```

```

const base = char >= 'a' ? 97 : 65;
result += String.fromCharCode(((code - base + shift) %
    <math>\rightarrow</math> 26) + base);
} else {
  result += char;
}
}
document.getElementById('caesar-result').innerText =
  'Encrypted
    <math>\rightarrow</math> Text: ${result}';
}
  
```

Next is the code for *Monoalphabetic Cipher Algorithm*. Monoalphabetic substitution encrypts text by replacing each character with another fixed letter based on a substitution key

```

function encryptMono() {
  const input = document.getElementById('mono-input').value;
  const key =
    <math>\rightarrow</math> document.getElementById('mono-key').value.toUpperCase();
  const alphabet =
    <math>\rightarrow</math> 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';
  let result = "";
  for (let char of input.toUpperCase()) {
    result +=
      <math>\rightarrow</math> alphabet.includes(char) ?
      <math>\rightarrow</math> key[alphabet.indexOf(char)] : char;
  }
  document.getElementById('mono-result').innerText =
    'Encrypted
      <math>\rightarrow</math> Text: ${result}';
}
  
```

Next is the code for *Vigenère Cipher Algorithm*. The Vigenère Cipher shifts plain- text letters using a repeating keyword.

```

function encryptVigenere() {
  const plaintext =
    <math>\rightarrow</math> document.getElementById("vigenere-plaintext").value.toUpperCase();
  const key =
    <math>\rightarrow</math> document.getElementById("vigenere-key").value.toUpperCase();
  let result = "";
  let keyIndex = 0;
  const alphabet =
    <math>\rightarrow</math> "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
  for (let char of plaintext) {
    if (alphabet.includes(char)) {
      const charIndex = alphabet.indexOf(char);
      const keyCharIndex = alphabet.indexOf(key[keyIndex %
        <math>\rightarrow</math> key.length]);
      const encryptedIndex = (charIndex + keyCharIndex) % 26;
      result += alphabet[encryptedIndex];
      keyIndex++;
    } else {
      result += char;
    }
  }
  document.getElementById("vigenere-result").innerText =
    <math>\rightarrow</math> 'Encrypted Text: ${result}';
}
  
```

Next is the code for *ubstitution Cipher Interactive Simulator*.

The Substitution Cipher simulator dynamically generates a randomized key for encryption.

```
const alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
let substitutionKey = alphabet.split('').sort(() => Math.random() - 0.5).join('');

document.getElementById("randomize").addEventListener("click",
  function () {
    substitutionKey = alphabet.split('').sort(() => Math.random() - 0.5).join('');
    displayKey();
  });

function encryptSubstitution(text) {
  return text.toUpperCase().split('').map(char => {
    let index = alphabet.indexOf(char);
    return index !== -1 ? substitutionKey[index] : char;
  }).join('');
}

function decryptSubstitution(text) {
  return text.toUpperCase().split('').map(char => {
    let index = substitutionKey.indexOf(char);
    return index !== -1 ? alphabet[index] : char;
  }).join('');
}
```

Next is the code for *Dynamic Visualization: Caesar Cipher*. This function updates the shifted alphabet dynamically based on user input from a slide

```
document.getElementById("shift-slider").addEventListener("input",
  function () {
    let shift = parseInt(this.value);
    let originalAlphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    let shiftedAlphabet = originalAlphabet
      .split('')
      .map((char, index) => originalAlphabet[(index + shift) % 26])
      .join('');
    document.getElementById("shifted-alphabet").innerText = shiftedAlphabet;
  });
```

Performance optimization was also important. Cipher algorithms were optimized for speed and efficiency to ensure real-time interaction without delays. The application was tested thoroughly on multiple browsers and devices, including mobiles and tablets, ensuring consistent performance everywhere.

Continuous feedback from usability tests guided improvements. Observations of user behavior, navigation paths, and errors provided valuable insights for refining the interface and functionalities. Code reviews helped improve code quality and ensured that encryption methods were reliable and efficient.

Finally, ethical considerations were strictly followed. Although no personal user data was stored, clear consent was obtained from participants during usability testing, explicitly informing

them about data use and protection measures. The details about testers, such as their educational background, were clearly described, ensuring transparency.

Overall, the methodological approach integrated adaptive design, ongoing user feedback, accessibility guidelines, and robust performance standards. This approach created an educational tool that made cryptography learning interactive, clear, and inclusive.

## 3.2 Usability Evaluation and Testing Methods

Evaluation combined user interaction analysis with peer code reviews. Test participants performed encryption tasks, explored tutorials, and completed quizzes. Observations revealed where learners struggled, how quickly they recovered from errors, and whether visual aids improved understanding. Feedback from participants guided iterative refinements, such as clarifying substitution key requirements or streamlining navigation.

Code reviews ensured maintainability and performance. Peer reviewers suggested refining input validation functions, optimizing encryption routines, and reorganizing code for responsiveness. Structured questionnaires measured instruction clarity, interface aesthetics, and system effectiveness. Results aligned with existing research that emphasizes iterative improvement, adaptive triggers, and inclusive design in educational interfaces [11, 12, 19].

## 4. USER EVALUATION AND EMPIRICAL ANALYSIS

### 4.1 Evaluation Framework and Methodology

- **User Interaction and Engagement Analysis:** This method involved observing users while they interacted with the application to assess usability, effectiveness, and user satisfaction. Some key findings included users facing difficulty entering correct shift values for the Caesar Cipher. Additionally, the guidance provided for the Monoalphabetic Cipher substitution keys was initially unclear, prompting improvements to simplify instructions. Testers' immediate feedback highlighted issues such as ambiguous labels and slow responses in certain browsers, which were addressed in later updates. Efforts were made to simplify error messages and provide clear instructions, resulting in noticeable improvements, especially in tasks involving interactive animations and quizzes.
- **Technical Review and Code Optimization:** A structured code review by experienced developers identified important issues, such as the initial Caesar Cipher implementation failing to handle special characters properly, leading to incorrect outputs. The Quiz Module initially lacked validation for blank answers, allowing incomplete submissions. Best practices were emphasized, including using modular JavaScript functions and consistent code formatting to enhance readability. Peer discussions led to optimization of the Vigenère Cipher algorithm for handling large texts efficiently. Additionally, better documentation with clear comments and explanations improved code maintainability.

## 4.2 User Feedback and Performance Metrics

Feedback was collected from participants through questionnaires and direct interactions, covering the following areas clearly:

- **Encryption Process Evaluation:** Participants rated their experience on a scale from 1 (Very Difficult) to 5 (Very Easy). They answered questions about the simplicity of encryption steps, clarity of input fields (messages and shift values), and any difficulties encountered during input. Participants also evaluated whether encrypted outputs matched their expectations and noted any issues or errors.
- **Decryption Process Evaluation:** Similarly, participants evaluated the decryption functionalities. Questions covered ease of understanding the decryption steps, clarity and usability of input fields for ciphertext and shift values, and issues encountered during inputs. Users assessed if decrypted outputs correctly matched the original plaintext and reported any difficulties.
- **Application Design Feedback:** Participants provided feedback on visual appeal, ease of navigation, and overall user experience. They evaluated whether the layout was visually attractive and comfortable, if buttons and interactive elements were easily identifiable, and if instructions and labels were clear. They also assessed whether the application clearly guided them through the encryption and decryption processes, providing an intuitive user experience.

## 4.3 Operational Challenges and Ethical Considerations

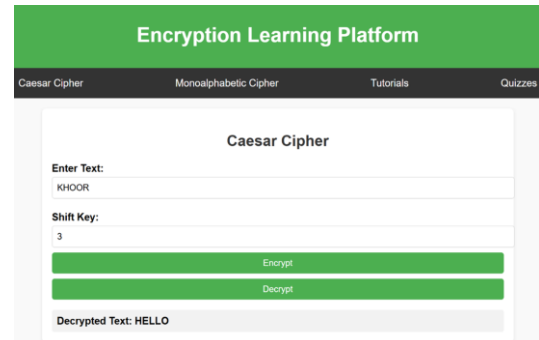
Operational challenges involved user input validation and managing errors effectively. Initially, non-numeric shift values caused errors in the Caesar Cipher, and substitution keys in the Monoalphabetic Cipher often had duplicates or missing characters. These were addressed by improving input validation and error messages to clearly indicate the issues.

## 4.4 Empirical Testing and Case Studies

- **Caesar Cipher Testing:** The Caesar Cipher allows users to encrypt and decrypt text by shifting letters by a specified value. During encryption, users input plaintext (e.g., "HELLO") and a shift value (e.g., 3), resulting in the ciphertext (e.g., "KHOOR"). Decryption reverses this process, recovering the original plaintext. Both processes were successfully tested. An example of the Caesar Cipher interface, showing encryption and decryption processes, is presented in Figure 2 and Figure 3.
- **Monoalphabetic Cipher Testing:** The Monoalphabetic Cipher substitutes each character in the plaintext with another character based on a user-provided substitution key. During encryption, users entered plaintext (e.g., "HELLO") and a substitution key (e.g., "QWERTYUIOPLKJHGFDSA ZXCVBNM"), producing the cipher-text (e.g., "ITKKG"). Decryption uses the same key to restore the original plaintext. An example of the Monoalphabetic Cipher Testing is presented in Figure 4 and Figure 5.
- **Vigenère Cipher Educational Tool:** The Vigenère Cipher encrypts plaintext by combining it with a keyword using modular arithmetic. During encryption, users entered plaintext (e.g., "HELLO") and a keyword (e.g., "SECURITY"), producing an encrypted output. Decryption used the same keyword to restore the original

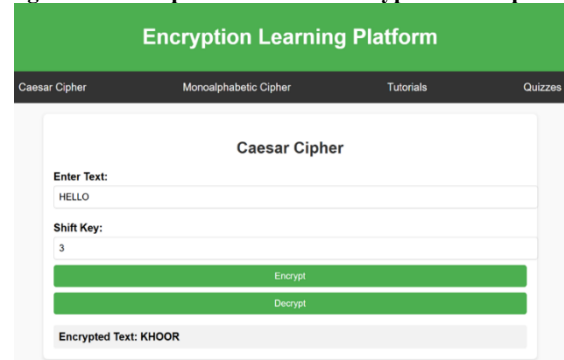
plaintext. An example of the Vigenère Cipher Educational Tool Testing is presented in Figure 6 and Figure 7.

- **Substitution Cipher Simulator Testing:** The Substitution Cipher Simulator allows users to experiment with plaintext-to-ciphertext transformations dynamically. During testing, users entered plain-text into the input field, and the ciphertext output was updated in real time. An example of the Substitution Cipher Simulator Testing is presented in Figure 8.



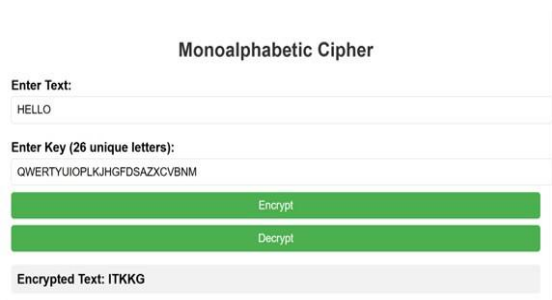
The screenshot shows the 'Encryption Learning Platform' interface for the Caesar Cipher. It has a green header with the platform name and a dark navigation bar with links to 'Caesar Cipher', 'Monoalphabetic Cipher', 'Tutorials', and 'Quizzes'. The main content area is titled 'Caesar Cipher' and contains an 'Enter Text:' field with 'KHOOR', a 'Shift Key:' field with '3', and two green buttons labeled 'Encrypt' and 'Decrypt'. Below these buttons, the 'Decrypted Text:' is shown as 'HELLO'.

Fig 2: Caesar Cipher Interface: Encryption Example



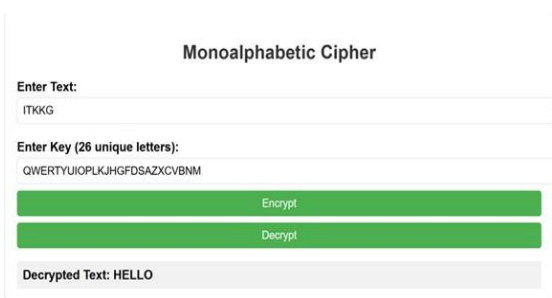
This screenshot shows the same 'Caesar Cipher' interface. The 'Enter Text:' field now contains 'HELLO', and the 'Shift Key:' field still has '3'. The 'Encrypt' and 'Decrypt' buttons are present. The 'Encrypted Text:' field at the bottom now displays 'KHOOR'.

Fig 3: Caesar Cipher Interface: Decryption Example



The screenshot displays the 'Monoalphabetic Cipher' interface. It features a green header and a dark navigation bar. The main area is titled 'Monoalphabetic Cipher' and includes an 'Enter Text:' field with 'HELLO', an 'Enter Key (26 unique letters):' field with 'QWERTYUIOPLKJHGFDSA ZXCVBNM', and 'Encrypt' and 'Decrypt' buttons. The 'Encrypted Text:' field shows 'ITKKG'.

Fig 4: Monoalphabetic Cipher Interface: Encryption Example



This screenshot shows the 'Monoalphabetic Cipher' interface with the 'Decrypt' button selected. The 'Enter Text:' field contains 'ITKKG', and the 'Decrypted Text:' field at the bottom displays 'HELLO'.

Fig 5: Monoalphabetic Cipher Interface: Decryption Example

**Vigenère Cipher Educational Tool**

**Practice Vigenère Cipher**

Enter Plaintext:  
HELLO

Enter Encryption Key:  
SECURITY

Encrypt

Decrypt

Encrypted Text: ZINFF

Fig 6: Vigenère Cipher Interface: Encryption Example

**Vigenère Cipher Educational Tool**

**Practice Vigenère Cipher**

Enter Plaintext:  
ZINFF

Enter Encryption Key:  
SECURITY

Encrypt

Decrypt

Decrypted Text: HELLO

Fig 7: Vigenère Cipher Interface: Decryption Example

**Substitution Table**

A	B	C	D	E	F
→ A	→ B	→ S	→ T	→ R	→ G
G	H	I	J	K	L
→ Y	→ P	→ J	→ Z	→ Q	→ O
M	N	O	P	Q	R
→ F	→ N	→ M	→ L	→ X	→ W
S	T	U	V	W	X
→ K	→ D	→ U	→ H	→ I	→ V
Y	Z				
→ C	→ E				

Randomize Key

Reset Key

**Input and Output**

Plaintext:  
A

Ciphertext:  
A

Fig 8: Substitution Cipher Simulator Example

- *Testing the Learning Hub - Basics of Cryptography:* Upon loading the "Learning Hub: Basics of Cryptography," users are presented with an introductory interface that features a visually appealing card layout designed to enhance readability. This interface includes a section titled "What is Cryptography?" which explains the purpose and significance of cryptography. Additionally, a "Types of Ciphers" section categorizes encryption techniques into

subsections for Caesar Cipher, Monoalphabetic Cipher, Substitution Cipher, and Transposition Cipher, providing a clear and organized overview. Interactive navigation menu links, such as "Introduction," "Types of Ciphers," "Interactive Animation," and "Quiz," enable users to seamlessly scroll to corresponding sections within the page. This design eliminates the need for reloading the page and ensures an organized progression of information. Each section is highlighted dynamically as users navigate, improving engagement and enhancing the learning experience. An example of Learning Hub Testing is presented in Figure 9.

**Learning Hub: Basics of Cryptography**

Introduction Types of Ciphers Interactive Animation Quiz

**What is Cryptography?**

Cryptography is the practice of securing information and communication through the use of codes, so that only those for whom the information is intended can read and process it. It plays a critical role in data security by ensuring confidentiality, integrity, and authentication.

**Types of Ciphers**

- Caesar Cipher: A substitution cipher that shifts characters by a fixed number of positions.
- Monoalphabetic Cipher: A cipher that substitutes each letter with another letter.
- Substitution Cipher: A more general form of replacing characters with other characters or characters.
- Transposition Cipher: A cipher that rearranges the order of the characters in a different order.

**Interactive Animation: Caesar Cipher**

Drag the slider below to rotate the Caesar Cipher with reference to the plaintext.

0 100

Original Message: HELLO WORLD! ROTATION: 3

Encrypted Message: RUHQGR ZRUOG!

Fig 9: Learning Hub: Introductory Interface

- *Testing the Quiz Module:* The Quiz Module enables users to assess their understanding of cryptographic concepts through multiple-choice questions. Users interact with the system by selecting answers to questions such as "What does cryptography ensure?" and "Which cipher rearranges the letters of plaintext?" Each question is accompanied by carefully designed answer options to challenge users' comprehension of key topics. Upon clicking the "Submit" button, the system evaluates the responses and provides immediate feedback. Correct answers are highlighted to affirm understanding, while incorrect answers are paired with detailed explanations or helpful hints to guide users toward the correct solution. This feedback mechanism reinforces learning by addressing misconceptions and encouraging users to refine their knowledge. An example of Quiz Testing module is presented in Figure 10.

### Test Your Knowledge: Cryptography Quiz

Answer the following questions:

Q1: What does cryptography ensure?

- ☐ Only Confidentiality
- ☒ Confidentiality, Integrity, and Authentication
- ☐ Only Integrity

Q2: Which cipher rearranges the letters of plaintext?

- ☐ Substitution Cipher
- ☒ Transposition Cipher
- ☐ Caesar Cipher

Submit

Your Score: 2/2

Fig 10: Quiz Module: Feedback after Submission

## 4.5 Usability Testing Outcomes

Table 1 and table 2 display the averages of the data collected from 10 subjects.



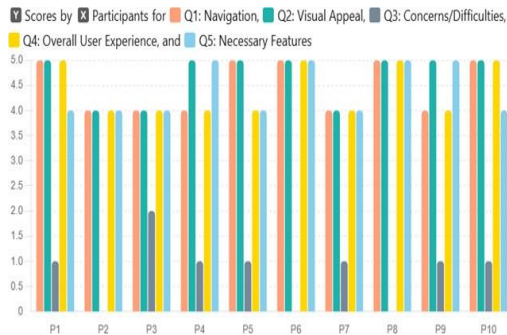
Figure 11 and figure 12 demonstrate individual results of each subject.

**Table 1: Evaluation Form 1 Results**

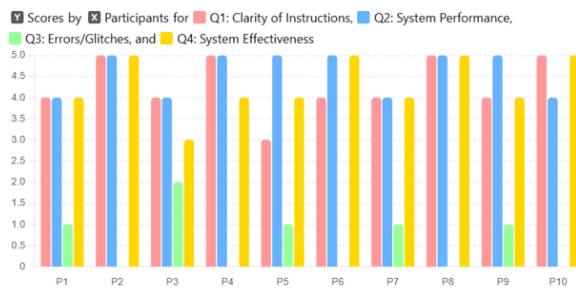
Participant	Navigation	Visual Appeal	Difficulties	User Experience	Features
<b>Average</b>	4.5	4.7	0.9	4.4	4.4

**Table 2: Evaluation Form 2 Results**

Participant	Clarity of Instructions	Performance	Errors/Glitches	Effectiveness
<b>Average</b>	4.3	4.6	0.6	4.3



**Fig 11: Average Ratings for Evaluation Form 1**



**Fig 12: Average Ratings for Evaluation Form 2**

## 5. CONCLUSION

This research showed clearly that adaptive user interfaces significantly enhance learning experiences in basic encryption methods. Interactive tutorials, immediate feedback, and adaptive hints effectively increased user engagement and simplified complex concepts. By combining adaptive UI generation [4, 5], user modeling [9], cultural customization [2], and accessibility considerations [3], the developed platform provides a practical foundation for inclusive and interactive cryptography education.

Future improvements can involve adding more advanced cryptographic methods, testing the application with a larger and diverse user group, and implementing machine learning techniques to better predict user interactions. Expanding multilingual support and culturally relevant examples could also enhance global accessibility and engagement. These steps would further improve the application's adaptability and overall effectiveness [10, 19, 20].

This research highlights how adaptive user interfaces can make learning basic encryption techniques easier, more engaging, and accessible to diverse learners. By using interactive visualizations, personalized feedback, and adaptive tutorials, the platform transforms abstract concepts into clear

and understandable lessons. The study shows that careful attention to usability, accessibility, and user engagement significantly enhances learners' experiences, even with challenging topics like cryptography.

With continued development, adaptive learning interfaces have great potential in education. By integrating user-centered design principles, cultural adaptability, and accessibility standards, this research provides a practical model that can be adapted for various subjects beyond cryptography. Ultimately, the strategic use of adaptive features, such as interactive tutorials and quizzes, sets a foundation for future digital educational tools, promoting effective and inclusive learning experiences.

## 6. REFERENCES

- [1] Peter Brusilovsky and Eva Millán. 2007. User Models for Adaptive Hypermedia and Adaptive Educational Systems. In *The Adaptive Web: Methods and Strategies of Web Personalization*. Springer, 3–53. [https://doi.org/10.1007/978-3-540-72079-9\\_1](https://doi.org/10.1007/978-3-540-72079-9_1)
- [2] Ayodele O. Daniel, Yinka Adio, Frank Isaac, and Stephen Adesina. 2013. Culture- Based Adaptive Web Design. *International Journal of Science and Engineering Research* 4, 2 (2013), 12–19.
- [3] Sergio Firmenich, Alejandra Garrido, Fabio Paternò, and Gustavo Rossi. 2019. User Interface Adaptation for Accessibility. (2019), 533–552. [https://doi.org/10.1007/978-1-4471-7440-0\\_29](https://doi.org/10.1007/978-1-4471-7440-0_29)
- [4] Krzysztof Z. Gajos, Daniel S. Weld, and Jacob O. Wobbrock. 2010. Automatically Generating Personalized User Interfaces With Supple. *Artificial Intelligence* 174, 12 (2010), 910–950. <https://doi.org/10.1016/j.artint.2010.05.005>
- [5] Beatriz Gamecho, Rafael Minón, Amaia Aizpurua, Iker Cearreta, Manuel Arrue, Nestor Garay-Vitoria, and Julio Abascal. 2015. Automatic Generation of Tailored Accessible User Interfaces for Ubiquitous Services. *IEEE Transactions on Human- Machine Systems* 45, 5 (2015), 612–623. <https://doi.org/10.1109/THMS.2015.2430305>
- [6] Giuseppe Ghiani, Marco Manca, and Fabio Paternò. 2015. Authoring Context- Dependent Cross-Device User Interfaces Based on Trigger-Action Rules. In *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia*. ACM, Linz, Austria, 313–322. <https://doi.org/10.1145/2836041.2836073>
- [7] Giuseppe Ghiani, Marco Manca, Fabio Paternò, and Carmen Santoro. 2017. Personalization of Context-Dependent Applications Through Trigger-Action Rules. *ACM Transactions on Computer-Human Interaction (TOCHI)* 24, 2 (2017), 14. <https://doi.org/10.1145/3057861>
- [8] Julio Guerrero-Garcia, Juan Manuel Gonzalez-Calleros, Jean Vanderdonckt, and Jaime Munoz-Arteaga. 2009. A Theoretical Survey of User Interface Description Languages: Preliminary Results. In *LA-WEB 2009: Latin American Web Conference*. Mérida, Yucatán, Mexico, 36–43. <http://www.usixml.org/en/guerrero-garcia-j-gonzalez-calleros-j-m-vanderdonckt-j-munoz-arteaga-j-a-theoretical-survey-of-user-interface-descriptio.html>
- [9] Dominikus Heckmann, Thomas Schwartz, Boris Brandherm, Michael Schmitz, and Michael von

- Wilamowitz-Moellendorff. 2005. GUMO—The General User Model Ontology. In *Proceedings of the International Conference on User Modeling*. Edinburgh, Scotland, United Kingdom, 428–432.
- [10] Johannes Helms, Ronny Schaefer, and Kris Luyten. 2009. Human-Centered Engineering of Interactive Systems With the User Interface Markup Language. In *Proceedings of the International Conference on Human-Centered Software Engineering*. Springer, Pisa, Italy, 139–171. [https://doi.org/10.1007/978-1-84800-907-3\\_7](https://doi.org/10.1007/978-1-84800-907-3_7)
- [11] Jamil Hussain, Waqas Anwar Khan, Muhammad Afzal, Mubashir Hussain, Bhargav H. Kang, and Sungyoung Lee. 2014. Adaptive User Interface and User Experience-Based Authoring Tool for Recommendation Systems. In *Proceedings of the International Conference on Ubiquitous Computing and Ambient Intelligence*. Springer, Belfast, United Kingdom, 136–142. [https://doi.org/10.1007/978-3-319-13102-3\\_24](https://doi.org/10.1007/978-3-319-13102-3_24)
- [12] Willemien Jorritsma, Frouke Cnossen, and Peter M. van Ooijen. 2015. Adaptive Support for User Interface Customization: A Study in Radiology. *International Journal of Human-Computer Studies* 77 (2015), 1–9. <https://doi.org/10.1016/j.ijhcs.2015.01.001>
- [13] Pat Langley. [n. d.]. Machine Learning for Adaptive User Interfaces. Intelligent Systems Laboratory, Daimler-Benz Research and Technology Center. <http://www.isle.org/~langley/papers/adapt.ki97.pdf>
- [14] Pat Langley. 1997. Machine Learning for Adaptive User Interfaces. In *Proceedings of the Annual German Conference on Artificial Intelligence*. Springer, Freiburg, Germany, 53–62.
- [15] Bettina Laugwitz, Theo Held, and Martin Schrepp. 2008. Construction and Evaluation of a User Experience Questionnaire. In *Proceedings of the Symposium of the Austrian HCI and Usability Engineering Group*. Springer, Graz, Austria, 63–76. [https://doi.org/10.1007/978-3-540-89350-9\\_6](https://doi.org/10.1007/978-3-540-89350-9_6)
- [16] Effie Law, Paul van Schaik, and Virpi Roto. 2014. Attitudes Towards User Experience (UX) Measurement. *International Journal of Human-Computer Studies* 72, 6 (2014), 526–541.
- [17] Quentin Limbourg, Jean Vanderdonckt, Benjamin Michotte, Laurent Bouillon, and Victor López-Jaquero. 2004. USIXML: A Language Supporting Multi-Path Development of User Interfaces. In *Proceedings of the International Workshop on Design, Specification, and Verification of Interactive Systems*. Springer, Hamburg, Germany, 200–220. [https://doi.org/10.1007/11431879\\_12](https://doi.org/10.1007/11431879_12)
- [18] Gerrit Meixner, Fabio Paternò, and Jean Vanderdonckt. 2011. Past, Present, and Future of Model-Based User Interface Development. *i-com* 10, 3 (2011), 2–11.
- [19] Najla Mezhoudi, Imane Khaddam, and Jean Vanderdonckt. 2015. Wisel: A Mixed Initiative Approach for Widget Selection. In *Proceedings of the 2015 Conference on Research in Adaptive and Convergent Systems*. ACM, Prague, Czech Republic, 349–356. <https://doi.org/10.1145/2811411.2811527>
- [20] Shrawan Shrestha, Praveesh Poudel, Sushant Adhikari, and Ishwor Adhikari. 2022. Adaptive Menu: A Review of Adaptive User Interface. *Trends in Computer Science and Information Technology* 7, 3 (2022), 103–106. <https://doi.org/10.17352/tcsit.000059>