



Edge Hill
University

Addressing the problem of insecure keyboard-derived patterns in user-generated passwords and PIN codes with a focus on smartphone users

Student Name: Veronika Chupova
ID number: 25868705

Supervisor: Dr Vinh-Thong Ta

28.09.2024

The statement: 'This Report is submitted in partial
fulfilment of the requirements for the MSc Computing
Degree at Edge Hill University.'

Originality Declaration

I, the undersigned, declare that the work presented in this research and development project is entirely my own, and that all sources of information have been acknowledged where appropriate. I understand that academic integrity is fundamental to maintaining the standards of this institution and that any form of academic dishonesty, including but not limited to plagiarism, collusion, or the use of unapproved external resources, is a serious violation of university policy.

I further confirm that:

- I have read and understood the institution's policies on plagiarism and academic misconduct.
- All direct quotations, ideas, data, and information derived from other works have been properly cited, and all paraphrased content has been appropriately referenced.
- This submission has not been previously submitted, in whole or in part, for any other course or degree.

I acknowledge that any breach of this originality statement may result in disciplinary actions in accordance with the university's regulations.

Student Name: Veronika Chupova

Student Number: 25868705

Signature: 

Date: 26.09.2024

Table of Contents

Abstract	7
Chapter 1. Introduction	8
1.1 Research Contributions	9
1.2 Ethics	10
Chapter 2. Literature Review	11
Chapter 3. Project Design & Management.....	16
Chapter 4. Implementation.....	17
4.1 Shuffling Algorithm.....	17
4.2 Password Policy	23
4.3 Software Requirements.....	24
4.4 System Architecture.....	24
4.5 Tools and Technology Stack.....	27
4.6 UI & UX Design.....	30
4.7 Keyboard Design.....	31
4.8 Software Development.....	33
4.9 Software Deployment.....	37
4.10 Data Collection.....	38
Chapter 5. Results Evaluation	39
5.1 Data Statistical Analysis	39
5.2 Passwords Sequential Pattern Analysis.....	42
5.3 PIN codes Sequential Pattern Analysis	46
Chapter 6. Conclusion	48
6.1 Research Limitations	49
6.2 Future Directions	49
References	51
Appendix A. Participant Consent Form - View (1/2)	57
Appendix A. Participant Consent Form - Text (2/2).....	58

Appendix B. Ethical Consideration List (1/3)..... 60

Appendix B. Ethical Consideration List (2/3)..... 61

Appendix B. Ethical Consideration List (3/3)..... 62

Table of Figures

Figure 1. Flowchart of the Fisher-Yates shuffling algorithm.....	18
Figure 2. Fisher-Yates shuffling algorithm visualisation	18
Figure 3. Flowchart of the Knuth shuffling algorithm.....	20
Figure 4. Knuth shuffling algorithm visualisation	20
Figure 5. Keyboard shuffling algorithm implementation with JavaScript	22
Figure 6. Sliding window algorithm visualisation	22
Figure 7. Shuffling validation algorithm.....	23
Figure 8. Multilayered architecture of experiment environment.....	27
Figure 9. Default on-screen keyboards of iOS and Android platforms.....	28
Figure 10. User interface design applied for experiment environment	31
Figure 11. Variation of default Android keyboard designs	32
Figure 12. Design of activated iOS keyboard button	35
Figure 13. Design of activated shuffled keyboard button: A – general view; B – implementation details	36
Figure 14. Schema of the collected data	39
Figure 15. Source platform analysis.....	39
Figure 16. Experiment completion time analysis	40
Figure 17. Experiment actions time analysis.....	41
Figure 19. Password input time and password length relation	42
Figure 20. Standard layouts of QWERTY keyboard and numpad.....	43
Figure 21. Translation algorithm for keyboard positional sequential patterns.....	44
Figure 22. Numboard positional sequential patterns visualisation	47

Table of Tables

Table 1. Comparison of shuffling algorithms 21

Table 2. Sequential strings’ strength comparison 45

Table 3. Passwords strength 46

Abstract

As online services become integral to daily life, the importance of robust data protection mechanisms, particularly password security, is undeniably critical. Despite early predictions of the refusal of passwords, they remain a primary line of defence against unauthorised access. However, the inherent simplicity of passwords often leads to vulnerabilities, as users frequently employ weak or repetitive combinations that are susceptible to cyberattacks. This study addresses the challenge of keyboard-derived patterns in passwords and PIN codes, particularly among smartphone users, through the innovative application of a keyboard shuffling algorithm. The research aims to enhance password strength by eliminating predictable sequences inherent in standard keyboard layouts. A mixed-methods approach was employed, combining qualitative and quantitative data collection, including a literature review of existing methodologies, algorithm development, and experimental validation. The findings reveal that the shuffled keyboard significantly mitigates the use of common patterns, resulting in stronger passwords and PIN codes as assessed by established password strength meters, including NIST entropy, Bitwarden and zxcvbn test. This research contributes to the field of cybersecurity by offering insights into user behaviour regarding password creation and by demonstrating the potential of shuffled input methods to improve security against prevalent vulnerabilities related to traditional keyboard layouts.

Chapter 1. Introduction

The more tasks people entrust to digital services, the more important data protection mechanisms become. Securing access to important essences with passwords proved a reliable, handy tool long before computer invention. Tough, according to Forbes (2022), Bill Gates, the founder of Microsoft, back in 2004 announced an imminent abandonment of password concept, it remains the frontier privacy shield up till now. However, the simplicity of this method goes hand in hand with its vulnerability, as passwords might be leaked, guessed, and cracked by attackers. Unfortunately, cybercriminals often have the upper hand, having the whole Internet to trawl through in search of poorly protected assets. Software developers do their best to enhance the protection of users' data by encrypting sensitive information, using multifactor authentication, implementing biometrical identification, and so on. However, if a secret string set by a user is initially weak, there is a greater threat it would be cracked at some point. Thus, ensuring robust passwords is the dominant interest for both sides in user-application relations.

Despite the constellation of password managers - special software designed to auto-generate secure passwords, only a minor portion of users accept the suggested variants, as highlighted by Ng et al (2022). Steves et al (2014) discovered that such behaviour is typically motivated by the need to remember the given passcode and re-use it from various access points. Therefore, the trade-off between the secret string complexity and memorability is inevitable. Because of that reason, the insecure practice of re-using the same passwords on multiple independent applications is widely spread. To address these poor behavioural habits, modern applications employ restrictive password policies tuned to check and accept only strings that satisfy specified requirements. However, striking number of users have learned to use simple, memorable combinations of repetitive patterns in passwords, successfully passing acceptance criteria. According to Yang et al (2021) the most frequent patterns are character repetitions (e.g. "11111", "aaaaa") and keyboard sequences such as "qwerty", "zxcvbn", "12345", etc.

This study addresses the challenges stemming from the frequent usage of keyboard-derived patterns in passwords and PIN codes by utilising a keyboard shuffling approach. Due to the significant transition of online activity toward mobile devices and the natural constraints in altering the default physical keyboards on laptops and PCs, this research focuses exclusively on smartphone users. Thus, the primary aim of this

study is to enhance the strength of user-generated passwords and PIN codes on smartphones through the utilisation of a keyboard shuffling approach. To achieve the stated aim, the following objectives will be accomplished in this work:

Objective 1 Literature Review. To conduct a comprehensive literature review of existing algorithms and methodologies employed in the IT industry to increase the security of passwords and PIN codes.

Objective 2 Algorithm Development. To design and develop an innovative keyboard shuffling algorithm aimed at eliminating keyboard-derived patterns from password and PIN codes.

Objective 3 Experiment. To develop an interactive solution incorporating the keyboard shuffling algorithm, and carry out an experiment with a cohort of smartphone users.

Objective 4 Results Evaluation. To assess the enhancement in strength of passwords and PIN codes generated using the developed keyboard shuffling algorithm.

In this work, an empirical methodology was employed to accomplish the aim and objectives. For this study, a mixed-methods strategy, integrating both, qualitative and quantitative data collection was applied to obtain insights. The research findings primarily relied on the analysis of observations obtained during the experiment.

1.1 Research Contributions

This study contributes to the field of cybersecurity by examining the effectiveness of a shuffled keyboard approach in eliminating frequently used patterns and, therefore, improving the strength of user-generated passwords and PIN codes. By focusing on patterns derived from the standard QWERTY keyboard and traditional numboard, the research provides valuable insights into the vulnerabilities associated with common password-creation practices.

One of the key contributions of this study is the development of the keyboard shuffling algorithm ensuring sufficient derangement that excludes character formations inherited from the QWERTY keyboard and standard numboard. The application of this algorithm in the task of password and PIN code creation ensured the absence of frequent keyboard-derived sequences in the collected data. Furthermore, the poor practice of providing keyboard sequential characters as a password or PIN code

resulted in strong strings when the shuffled keyboard was used. The respective assessment was performed using widely recognized Password Strength Meters, including US National Institute of Standards and Technology (NIST) entropy, Bitwarden, and zxcvbn test.

Overall, this research contributes to a deeper understanding of password and PIN code strength by demonstrating the high potential of shuffled input methods to reduce vulnerabilities linked to traditional keyboard layouts.

1.2 Ethics

Given this study focuses on passwords and PIN codes which are sensitive information, strict alignment with existing laws, ethical standards, and social norms was ensured throughout all project stages. The idea guiding this work is to serve the greater good of society by upholding principles of integrity, accountability, honesty, safety, and excellence.

The study was conducted under the University's supervision and concerned the academic community. All uses of intellectual property and other materials were handled with respect for copyright laws and properly referenced, with contributions to the research project acknowledged and publicly credited.

Before involving third-party resources—such as web-hosting providers, programming tools, frameworks, cryptography, and applications— potential legal, ethical, and reputational implications were thoroughly assessed.

The research required the participation of smartphone users who interacted with the developed software. To ensure informed consent, each participant received a detailed consent form (Appendix A) that outlined the research purposes, experimental plan, types of data collected, and consent withdrawal policy. The consent form also included a clear warning advising participants against using their current or previous passwords and PIN codes in the study.

To minimise risks associated with sensitive data, the software and experiment design excluded the collection of any information that could identify users.

The set of measures was implemented towards the requirements and guidelines included in the following legal acts:

- Edge Hill Research Ethics Policy and the Research Data Management Policy. The principles outlined in these documents guided the research design. The principles outlined in these documents guided the research design. Any study-related activities were discussed and endorsed by the University's ethical committee. Ethical consideration list can be found in Appendix B of this report.
- General Data Protection Regulation (GDPR) (2018). Only the data necessary for the research was collected within the experiment. A lawful basis for data processing was established by obtaining participants' informed consent. The consent form was developed in adherence to the principle of transparency, clearly stating participant rights and data security measures, and explicitly demonstrating research interest.
- UK Consumer Rights Act (CMA) (2015). This study was designed to respect participants' right to fair treatment, avoiding deceptive practices and unsafe requirements. The liabilities and responsibilities subsequent to the failure of this legal act were considered and taken into account.
- UK Data Protection Act (DPA) (2018). To comply with this document, necessary actions were taken to ensure research accountability by keeping detailed records of data processing activities and conducting regular reviews of data handling practices.

Chapter 2. Literature Review

With the growth of digital assets concealed behind passwords the number of cybercriminals drawn to get unauthorised access to them increases accordingly. Offenders can breach security in several ways. They might directly obtain passwords by peeking over someone's shoulder, eavesdropping, stealing recorded data, or using psychological tricks and scams. Additionally, they may access information indirectly through cyber attacks. Yu and Huang (2015) subdivide the latter into three types:

- Rainbow table attack. In this scenario, criminals attempt to recover passwords from their encrypted or hashed form obtained from database leakages;
- Dictionary attack. This type involves hackers utilising a pre-compiled list of stings, substrings and their combinations commonly used as passwords;
- Brute force attack. This scenario implies continuous systematical testing of all possible combinations of symbols to discover the correct password.

To address the threat of database leaks exposing passwords encryption methods have been extensively used for many years. Several recent studies have focused on improving password encryption as a standard method for protecting data storage in modern digital solutions. Lustro (2019) effectively enhanced the secureness of passwords encrypted with the widely used BCrypt algorithm. They implemented a mechanism for dynamic salt generation (an arbitrary string of characters added to a user-generated password), ensuring that the resulting hash was consistently unique, which was recognised as an effective defence against rainbow table attacks. Similarly, Senthilselvi et al (2024) sought to minimise the probability of hash collisions for identical passwords. Unlike Lustro (2019), they enhanced hash robustness by randomly chaining several asymmetric cryptography algorithms, including SHA256, SHA384, SHA224, SHA512, SHA1, MD5, and RIPEMD160. However, the performance of the proposed method was not clearly outlined, raising concerns about its practical applicability, given that encryption algorithms are intentionally designed to be computationally intensive and time-consuming to obstruct brute force attacks.

Despite the admitted effectiveness of traditional asymmetric encryption methods highlighted by Lustro (2019) and Senthilselvi et al (2024), Agarwal et al (2021) argued that all such algorithms are ultimately susceptible to brute force attacks. The vulnerability arises from the fact that each attempt to decrypt ciphered text with the incorrect key results in nonsensical output, which can be easily distinguished from genuine information, guiding attackers to continue their efforts. In response, they proposed an innovative cryptographic primitive named "Honey Encryption". This approach promises effective defence against offline brute force attacks by employing an algorithm for generating hashes that produce plausible outcomes, resembling legitimate passwords in the result of every decryption attempt.

Since most brute force attacks are conducted offline due to the significant computational power required and the constraints imposed by network rate limits, the honey encryption method proposed by Agarwal et al (2021) appears more promising than the straightforward ciphering of high-complexity suggested in the studies of Lustro (2019) and Senthilselvi et al (2024). Additionally, the findings of Ntantogian et al (2019) indicate that increasing encryption complexity can make a system more vulnerable to service denial attacks. This dependence significantly limits approaches that rely heavily on encryption and hash complexity for password protection, dictating the need for additional mechanisms to strengthen password security.

In addition to rainbow table and brute force attacks, dictionary attacks are commonly used in password-guessing attempts. Most of these attacks are designed to trawl through and test previously leaked password databases, according to Carnavalet and Mannan (2015). Their study also highlights that strong passwords can significantly reduce the risk of being compromised. In this context, several recent studies have focused on methodologies for assessing password strength, leading to their widespread use in password management systems. One notable example is the innovative research by Thanh and Tanaka (2024) which proposed a new metric for evaluating the security of user-generated passwords. The authors enhanced previously used entropy-based methods by incorporating statistical analysis of the self-information contained in the analysed strings. The study demonstrated how previously leaked passwords can be beneficial for developing effective protection mechanisms. On the other hand, while the proposed algorithm outperformed the accuracy of earlier models in evaluating password strength, it did not consider password memorability—a critical factor for sustainable user-application interactions, as highlighted by Rodriguez et al (2022). Additionally, successful attacks are not solely empowered by leaked databases. Kanta et al (2022) demonstrated that publicly accessible contextual information about users, supplemented by ontology graph data, can constitute a bespoke dictionary sufficient for targeted hacking.

In terms of attack performance, Alkhwaja et al (2023) compared brute force and dictionary cracking methods, demonstrating that the latter outperforms the former in computational power and time required, even when brute force scripts run on a parallel computing environment with advanced hardware. Another significant finding from Alkhwaja et al (2023) revealed that synthetically generated passwords are more resilient against dictionary attacks, thereby revisiting the confrontation between string complexity and memorability previously discussed by Rodriguez et al (2022).

This relative simplicity and cost-effectiveness of dictionary attacks may encourage hackers to attempt this method at the beginning of the password-cracking process each time. Given this context, maintaining sufficient password hygiene is crucial.

In this regard, Shi et al (2021) empirically demonstrated that most password-breaking offline algorithms are sensitive to various factors, including the length and structure of a secret string, a set and variation of symbols used, contextual user information, etc. In addition to finding that no single password cracker is universally effective, they also

discovered that the performance and efficiency of cracking vary among different password groups based on regional and linguistic characteristics.

Both, Alkhwaja et al (2023) and Shi et al (2021) agreed that password strength is a key factor that can deter hackers, given the significant resources required to breach strong ones. In contrast, Güven et al (2022) revealed that users often create relatively simple passwords that frequently include repetitive character sets and easily predictable sequences. A similar conclusion was drawn by private research published by Datagenetics.com (2012), which statistically analysed PIN code leaks.

To address this problem, researchers proposed various methods to help users create stronger passwords. For instance, Singh and Raj (2019) developed an algorithm that enables dynamic changes to password policies, making it difficult for attackers to predict the required characteristics of secret strings. Their method specifically aims to enhance user-generated passwords by incorporating rarely used symbols, such as !%\$#, into acceptance criteria. In contrast to the previously discussed works of Thanh and Tanaka (2024), Alkhwaja et al (2023), and Singh and Raj (2019), the research of Glory et al (2019) was focused on balancing password memorability with its strength. The authors proposed a novel algorithm for password generation, which produces strings that are highly resistant to brute force attacks. In their work, they combined user-related information to create strong passwords, relying on the assumption of user context uniqueness. However, Ng et al (2022) later revealed that user-generated passwords remain prevalent, with only 6% of users choosing the suggested secure passwords instead of their own.

Interestingly, studies by Güven et al (2022), Datagenetics.com (2012), and Statista.com (2019) indicate that over half of the commonly used passwords contain sequential substrings of symbols found either on a standard QWERTY keyboard or a numeric telephone-styled keypad. A report from Steves et al (2014) identified that the reasons for this behaviour include not only the simplicity of typing and memorising passwords, but also the convenience of passwords renewing by shifting a combination one character over. These well-known repetitive patterns are frequently exploited in dictionary attacks.

This persistent practice that people use to create their secret strings has attracted the attention of many researchers aiming to address keyboard-motivated vulnerabilities in password security. Chou et al. (2012) developed a framework to identify possible

keyboard positional patterns of the standard keyboard and simulated a new attack method based on these compiled patterns. Although their algorithm did not surpass dictionary attacks in password cracking, it significantly outperformed brute force attacks, increasing concerns about this vulnerability. Later, Wang et al. (2016) demonstrated that incorporating keyboard patterns into their attack model was effective not only for offline cracking but, more importantly, for online guessing, successfully bypassing protective throttling mechanisms of access systems.

In contrast to Chou et al. (2012) and Wang et al. (2016), Lyu et al. (2022) showed that the standard keyboard layout can also be used for good, enabling strong password creation. They introduced a novel mnemonic technique for secret string generating from keyboard characters located on the avoiding path around a chosen word. While the authors demonstrated the effectiveness of this method against attacks, it still risks creating vulnerable keyboard-derived pattern strings, particularly if a chosen word is formed within characters from a single keyboard row.

Since recognising the harmful effects of the standard keyboard layout on password security, little has been done to modify the keyset itself under the prevalence of desktop and laptop computers with physical keyboards. The initial attempts to secure passwords through keyboard shuffling emerged with the rapid increase in smartphone usage. Addressing password stealing through screen capturing and shoulder-surfing techniques, Agarwal et al. (2011) proposed a novel approach that involved dynamically shuffling the keyboard layout while concealing characters during touch events. Although their method significantly improved defences against password leakage, its usability remains questionable and the authors acknowledged the substantial time required to find each character with every new layout as well. Similarly, Brindha et al. (2018) applied keyboard shuffling techniques to strengthen Automated Teller Machine (ATM) security against the stealing of customers' PIN codes when the input is organised with a touchpad. Their method proved promising protection from both, on-spot visual shoulder-surfing and residue-based attacks that exploit heat and stain remnants from touchscreen use. Schneider et al. (2019) also emphasised the effectiveness of altering keyboard layouts for virtual reality (VR) users, where real-world surroundings are obscured, exposing the password entry process to others and, consequently, raising security concerns. However, the studies by Agarwal et al. (2011), Brindha et al. (2018), and Schneider et al. (2019) did not focus on the password creation process, nor did they attempt to apply their keyboard shuffling algorithms to enhance

password strength, instead concentrating solely on improving defences against password stealing.

To summarise, this focused literature review revealed the importance of strong passwords within the broader context of cybersecurity and highlighted the harmful effects of standard keyboard layout patterns if applied as secret strings. To address this challenge, this study aims to apply the keyboard shuffling method, extensively researched for password protection, to the password creation process in an attempt to enhance the strength of user-generated secret strings.

Chapter 3. Project Design & Management

Given the constraints of time and resources—whether human or material—implementing an effective project management (PM) framework is essential for achieving high-quality outcomes within the established period. An effective PM framework not only helps in organizing tasks and resources efficiently but also ensures that all project milestones are met in a timely manner, thereby enhancing overall productivity. Additionally, this research involves a stage of software development, which requires the use of an appropriate Software Development Life Cycle (SDLC) model. The SDLC provides a structured approach to the development process, guiding the project from initial concept to deployment and maintenance. In this study, the choice of both, a PM framework and an SDLC model was primarily driven by their complementarity.

In this study, two widely recognised project management frameworks were evaluated: Projects in Controlled Environments (PRINCE) and Project Management Body of Knowledge (PMBOK). The first one, known for its structured methodology, is suited well for large-scale projects that require strict control and management. As followed from the research by Volovyk and Harmash (2022) the PRINCE framework is the process-based linear methodology that ensures moving initiatives through defined pipeline, which makes this approach to project management cumbersome and less flexible. PMBOK, on the other hand, represents an abundant set of best practices, guidelines and recommendations, which can be flexibly compiled and utilised as necessary. Its plan-based methodology allows for guided adaptation throughout the project lifecycle, as highlighted by Karaman and Kurt (2015). Clear advantages of the PMBOK framework over PRINCE for the specifics of research projects motivated the choice of the former for this study.

Consequently, the choice of Software Development Life Cycle (SDLC) models was narrowed down to plan-driven ones to ensure seamless alignment with the employed PM framework.

Software development within the context of academic research presents several unique characteristics, such as precise and immutable requirements stemming from clear aims and objectives, thoughtfully structured workload, pre-defined evaluation processes, and a limited trustworthy workforce. These factors make the Waterfall model, the oldest sustainable SDLC framework, particularly suitable for this study. Its sequential nature allows for a clear extraction of stages, making it easy to integrate within the PMBOK framework.

This combined approach not only streamlines the software production process but also maintains comprehensive control over the entire project. Ultimately, this synergy between the project management and software development frameworks is crucial for overcoming the difficulties of academic research and achieving assigned objectives. In accordance with PMBOK guidelines, the research project scope was structured into five core process groups: initiating, planning, executing, monitoring & controlling, and closing. Each process group was further broken down into smaller, atomic tasks with clearly defined, measurable outcomes. Given the interdependencies between subtasks—when some tasks cannot proceed until others are completed—the overall project plan follows a staircase-like structure, except for the monitoring and controlling activities, which occur regularly throughout the project. This approach ensures a systematic and organized progression while maintaining flexibility for ongoing monitoring and amending.

Chapter 4. Implementation

4.1 Shuffling Algorithm

This section presents the informed design and development of a novel algorithm to shuffle the traditional QWERTY keyboard layout to address the issue of commonly used keyboard-derived patterns in user-generated passwords and PIN codes. To develop the algorithm ensuring sufficient shuffling of letter and number sequences, widely adopted methods were reviewed, and their performance on time and memory complexity was evaluated. The Fisher-Yates algorithm proposed in Aylmer (1938), is

one of the most well-known methods for deranging sequences. It works by selecting a random index from the initial range, removing the corresponding element from the original sequence, and placing it into the shuffled sequence. With each iteration, the range of available indices decreases by one, and the process continues until no elements remain in the original sequence. The algorithm's logical schema and visual representation are displayed in Figure 1 and Figure 2 respectively.

Figure 1. Flowchart of the Fisher-Yates shuffling algorithm

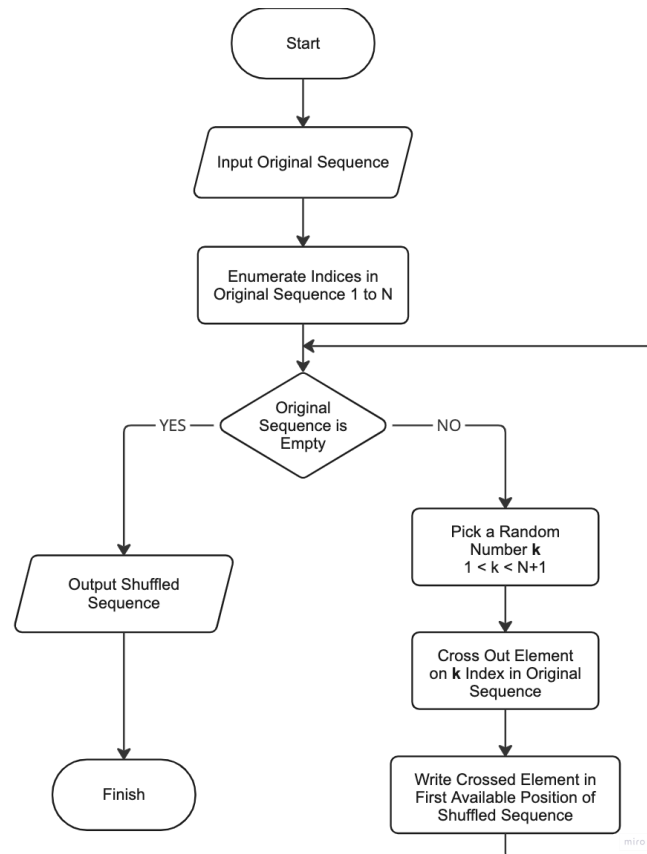
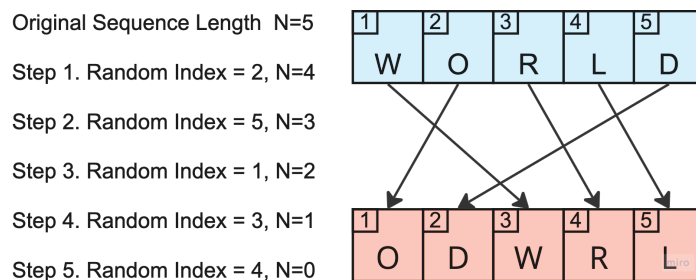


Figure 2. Fisher-Yates shuffling algorithm visualisation



Although this method was introduced before the computer era, it can still be implemented in any modern programming language. However, the time and memory complexities of the Fisher-Yates algorithm are not optimal. The process of selecting an element must be repeated n times, where n represents the number of elements in the original sequence. Each iteration involves removing the randomly selected element from the original array, which typically requires shifting the remaining elements—an operation with a time complexity of $O(n)$. As a result, the overall time complexity of the algorithm is $O(n^2)$, with a memory space complexity of $O(2n)$.

Later, Durstenfeld (1964) optimised the original Fisher-Yates algorithm and designed a version specifically suited for computer implementation. This improved method, often referred to as the Knuth shuffle, works by iterating backwards from the last element of the array and swapping it with an element at a randomly selected index from the unshuffled portion of the sequence, which starts at index 0 and ends at the current index. Since this algorithm only requires a single pass through the sequence, its time complexity is $O(n)$, where n is the number of elements. Additionally, because the Knuth shuffle allows for in-place array mutation, it achieves a memory space complexity of $O(n)$, reducing the memory requirements by half compared to the original Fisher-Yates algorithm. The algorithm's logical schema and visual representation are shown in Figure 3 and Figure 4 respectively.

Figure 3. Flowchart of the Knuth shuffling algorithm

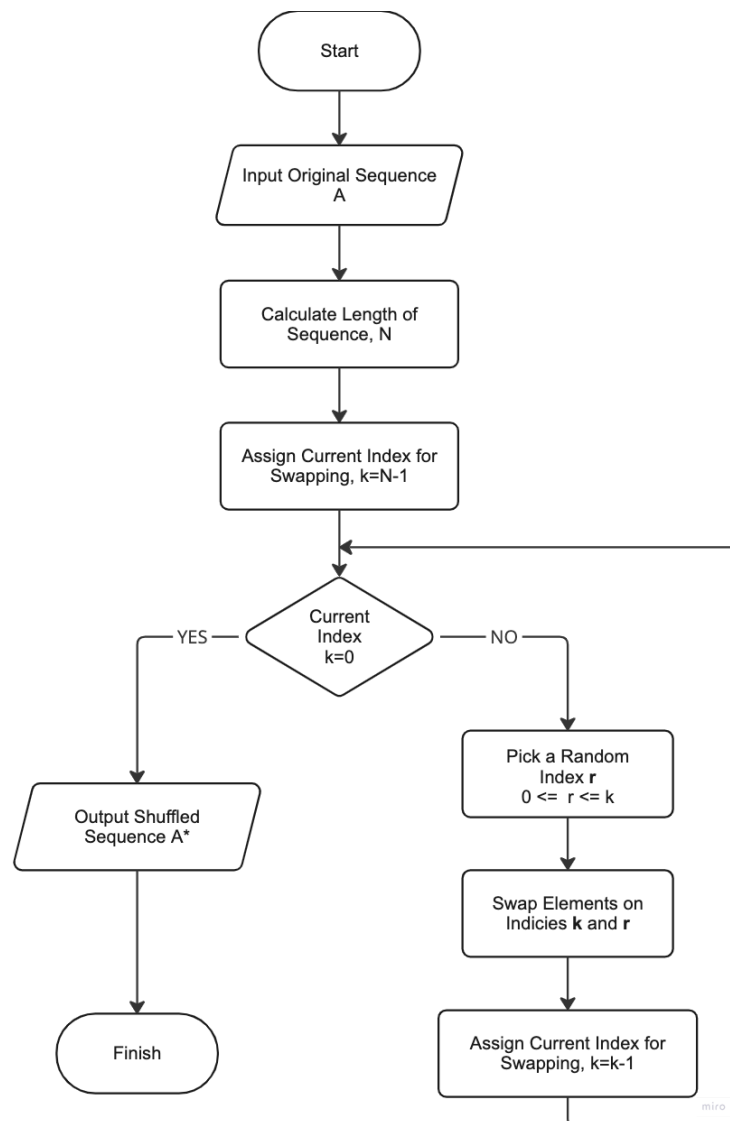
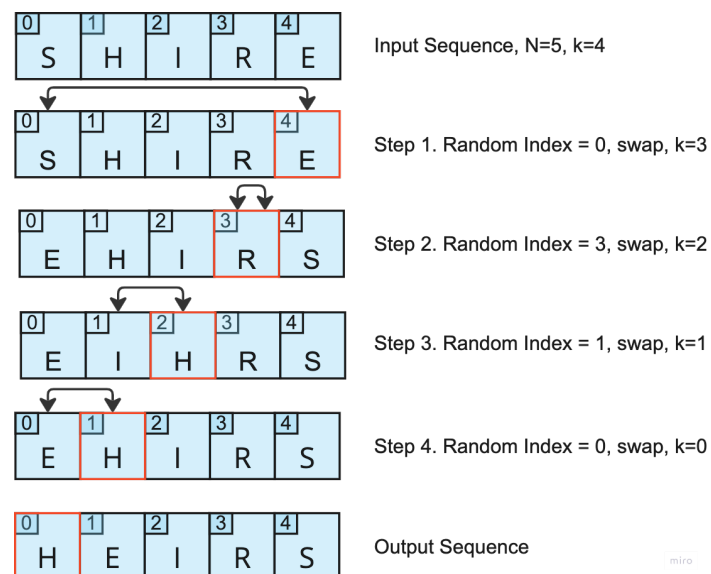


Figure 4. Knuth shuffling algorithm visualisation



Although the Fisher-Yates and Knuth algorithms produce uniformly shuffled sequences, some elements may remain in their original positions, meaning the resulting array is not a true cyclic permutation. Sattolo (1986) addressed this limitation by introducing a new variant of the Knuth algorithm. Like the Knuth method, the Sattolo algorithm iterates backwards through the array, but the range for selecting a random index excludes the current element being processed. This ensures that every element in the original sequence changes position by the end of the algorithm. Despite this modification, the algorithm retains the same time and memory complexities of $O(n)$ as the Knuth method.

Since the Sattolo algorithm guarantees a cyclic permutation while maintaining the lowest time and memory complexity as can be seen from Table 1, it was employed for shuffling keyboard sequences in this study.

Table 1. Comparison of shuffling algorithms

	Time Complexity	Memory Complexity	Cyclic Permutation
Fisher-Yates	$O(n^2)$	$O(2n)$	X
Knuth	$O(n)$	$O(n)$	X
Sattolo	$O(n)$	$O(n)$	V

Next, the original array which will be taken by the shuffling algorithm as input was designed. Although the traditional rows of the QWERTY keyboard and the standard numboard might be treated as separate arrays for shuffling, this approach was rejected. The primary concern was that keeping the same set of characters within each row would make it easier for users to recreate familiar line-based patterns and use them as passwords or PIN codes. Instead, the entire sequence of characters arranged either alphabetically for letters or in ascending order for numbers, was set as the input array for the shuffling algorithm. The algorithm implementation with JavaScript programming language is illustrated in Figure 5.

Figure 5. Keaboard shuffling algorithmn implementation with JavaScript


```
const array = [...Array(26).keys()].map(each => String.fromCharCode(each + 97))
function shuffle(array) {
  let remain = array.length
  let i
  while (remain > 0 ) {
    i = Math.floor(Math.random() * remain--);
    [array[i], array[remain]] = [array[remain], array[i]]
  }
  return array
}
```

Then, a validation algorithm was designed and implemented to assess the quality of the shuffling, ensuring sufficient derangement and the absence of line-based sequences longer than two characters from the standard keyboard. Yang et al (2021) identified and compiled the most common positional patterns used to create passwords, noting that horizontal sequences are the most frequent. These findings were applied in this study to develop an algorithm that analyses the shuffled array for sequences based on the standard keyboard layout. The core concept of this algorithm is a sliding window that moves through the shuffled sequence, searching for lines of three consecutive characters from the QWERTY keyboard. If the specified sequence is detected, the shuffling is admitted insufficient. In this case, the shuffling algorithm is rerun and repeated until the validation function confirms that the array is sufficiently randomised. The visual representation of this function is shown in Figure 6.


Figure 6. Sliding window algorithm visualisation

Example of Sliding Window Algorithm (window size = 6)


Step 1. Outcome forth sequence 'qwerty', reverse sequence 'ytrewq'




Step 2. Outcome forth sequence 'wertyu', reverse sequence 'uytrew'



Step 3. Outcome forth sequence 'ertyui', reverse sequence 'iuytre'



Step 4. Outcome forth sequence 'rtyuio', reverse sequence 'oiuytr'



It is important to note that, as highlighted by Yang et al. (2021), keyboard patterns in passwords include not only forward left-to-right sequences but also their reverse readings. Therefore, the validation algorithm was tuned to analyse the shuffled sequence for all possible line-based patterns of three characters, both forward and backward. The code snippet for this algorithm is displayed in Figure 7.

Figure 7. Shuffling validation algorithm

```
function quality_check(shuffled, refference) {
  const shuff_string = shuffled.join('')
  const result = refference.reduce((acc, line) => {
    for (let end = 2; end < line.length; end++) {
      const frame = line.slice(end-2, end+1)
      const reversed = frame.reverse()
      acc = acc || shuff_string.includes(frame.join('')) ||
shuff_string.includes(reversed.join(''))
    }
    return acc
  }, false)
  return result
}
```

4.2 Password Policy

The steady growth in computational power over recent decades has not only brought goods but also empowered the development of more sophisticated malicious software. In response, the complexity requirements for passwords, often referred to as password policies, increased accordingly. For instance, the industry standard for password length, set at a minimum of 8 characters by both the NIST (Grassi et al (2017) and the UK National Cyber Security Centre (NCSC) (2018), is widely adopted by developers. However, another widely recommended guideline—avoiding additional composition rules—is not as strictly enforced. Suood Alroomi and Li (2023) revealed that most examined websites implemented 2 or 3 restrictive criteria, such as requiring numbers, special symbols, or uppercase letters, in their password policies. To create an experimental environment closely mirroring real-world conditions and to accurately reflect participants' typical behaviour, this study adopted a password policy similar to that of most web applications. Therefore, the restrictive password requirements incorporated into the experiment facility are as follows:

- Minimum length of 8 characters
- At least 1 lowercase Latin letter
- At least 1 uppercase Latin letter
- At least 1 number

4.3 Software Requirements

Before proceeding with the software planning and development stages, a set of core system requirements, derived from the nature of the problem, was established as follows:

1. Mobile device compatibility. Since most personal computers and laptops use physical keyboards for typing, introducing a virtual keyboard with different layouts could confuse participants and lead to experiment failures. Therefore, the development focuses exclusively on mobile device users, where virtual keyboards are natural, allowing the developed keyboard to replace the default one during the experiment.
2. Platform independence. With a wide range of mobile devices in use, it is crucial to include as many as possible to avoid bias from platform-specific features or behaviours. Additionally, different hardware and software vendors have distinct application management approaches and technology stacks. Given the study's time constraints, the experimental setup must be compatible with a broad range of mobile devices to ensure convenience and inclusivity.
3. Ad-hoc User Interface (UI) support. To accurately assess the impact of the shuffled keyboard on user-generated passwords and PIN codes, participants should experience a familiar user flow, using well-recognised UI design patterns. The experimental setup should display the new shuffled virtual keyboard over the standard one while maintaining a seamless, habitual interaction experience. Therefore, a key requirement for the solution is the creation of a custom, responsive, and interactive interface.
4. Remote access. Considering the limited time available for the study and the need to gather a sufficient number of observations for statistical validity, the experimental platform must be accessible online to multiple users simultaneously.

4.4 System Architecture

The system architecture for the research facility was shaped by the need to meet the key requirements outlined in the previous section of this report. However, there are many architecture styles, component topologies, and software design patterns, each suited to specific software purposes depending on their objectives. To select an appropriate structure for the research facility, a classification of software architectures

was explored through three prospects: high-level organizational architecture, structural/deployment level, and implementation level, also known as design patterns. At a high level, software applications can be organized in the following ways:

- Unitary Architecture. According to Richards (2020), this was the earliest style of software organisation, used before the widespread of personal computers. Applications built with this architecture are self-contained, single-source programs typically executed as a single process. These systems face challenges with scalability, accessibility, and maintenance. However, many modern desktop applications, such as Microsoft Word and Calculator, continue to use this architecture style.
- Client/Server Architecture. The most commonly used structure for multi-user applications. Richards (2020) determined this architecture to be the most suitable for modern online-accessible software, which lacks the limitations of Unitary Architecture. In this model, software is divided into two distinct entities: the Client, which requests and applies resources, and the Server, which handles business logic, resource management, and navigation. Client/Server Architecture is widely used in web applications, email systems, file servers, and database systems.
- Peer-to-Peer Architecture. Often referred to as decentralised, this architecture lacks a unified server to manage client requests. According to Koegel et al (2009) each user, referred to as a Peer, is equipped with an application that can search for and interact with other Peers in the network. Each Peer acts as both, a Client when requesting resources and a Server when responding to requests from others. Examples of Peer-to-Peer architecture include platforms like BitTorrent and Bitcoin.

After evaluating the strengths and weaknesses of the three high-level architecture styles, the Client/Server Architecture was selected as the most suitable for meeting the key system requirements. A web application was chosen as the technological solution to express the developed algorithm and collect data for assessing its impact on user-generated passwords and PIN codes strength.

There are various approaches to building and deploying a web application built on Client/Server Architecture, depending on factors such as business logic complexity, component autonomy, data streams, expected workload, etc. While no two software

solutions are identical, at the Structural Architecture level, they generally fall into one of the following designs explored by Geoffrey et al (2024):

- 2-Tier Architecture: The simplest software structure, suitable for small to medium-sized applications with moderate security requirements. While it benefits from development simplicity and a homogeneous codebase, applications built with this approach are less scalable, less portable, and less secure compared to others.
- 3-Tier Architecture: This differs from the 2-tier model by adding an extra layer that encapsulates data access, enhancing system scalability, maintainability, and data security. However, the more distributed structure increases traffic, which may cause performance latency compared to other architectures.
- N-Tier Architecture: Typically used for large-scale, high-load systems that are often integrated with other self-contained software. This structure improves scalability, sustainability, and data integrity due to a high extent of logical and physical separation. While N-Tier architecture offers many advantages, it requires significant resources for development, deployment, and maintenance.

Considering the resource constraints and time limits of this study, the simplest 2-tier Architecture was considered an affordable and sufficient structure for developing the research software.

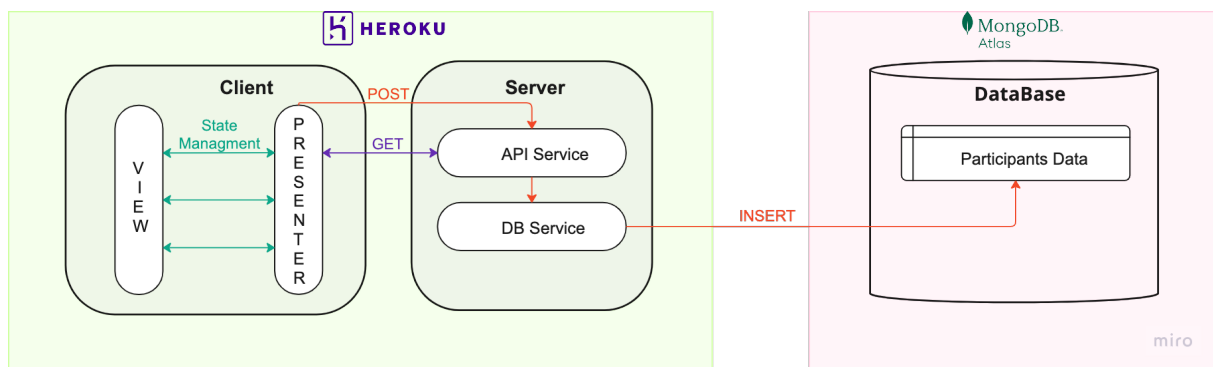
During the experimental phase of this study, participants will interact with the developed system in various ways, such as receiving information, typing using the shuffled keyboard, entering data, and submitting forms. These interactions require a complex user interface (UI), making it necessary to consider an additional abstract level of system organisation known as Architectural Design Pattern. For web applications, a Model-View (MV) framework is commonly used to structure the interactions between the front-end and back-end of the software. In this framework, a Model refers to the application's business logic, API endpoints, data management, and other backend operations, while a View represents the user interface displayed to participants. Jr (2015) proposed the following classification of Architectural Design Patterns:

- MV (Model-View): The simplest and most straightforward pattern, ideal for displaying data directly from a Model.

- MVC (Model-View-Controller): Suitable for more complex UIs that involve user interactions and state management. The Controller component handles input and updates the Model accordingly.
- MVP (Model-View-Presenter): Similar to MVC, but it ensures complete separation of business logic from the presentation layer. The Presenter takes on the role of the Controller while also updating the View according to the Model's state, ensuring no direct interaction between the Model and the View.
- MVVM (Model-View-ViewModel): A unique pattern that binds the UI elements of the View to an intermediate actor called ViewModel. This enables targeted updates to the View without the need for full re-rendering, ensuring low-latency UI updates.
- MVU (Model-View-Update): Unlike other patterns, MVU follows a declarative, functional approach. It transforms user inputs or events into corresponding model states and creates a new View based on the updated model.

Since the experiment in this study involves handling user-generated passwords and PIN codes, which are considered sensitive information, an architectural pattern that ensures the greatest separation of components and minimal data transfer is preferred. For this reason, the MVP pattern was selected for implementation in this work. The comprehensive analysis and evaluation of software architectures guided the selection of the most appropriate and practical structure for the experiment facility. The complete architecture is illustrated in Figure 8.

Figure 8. Multilayered architecture of experiment environment



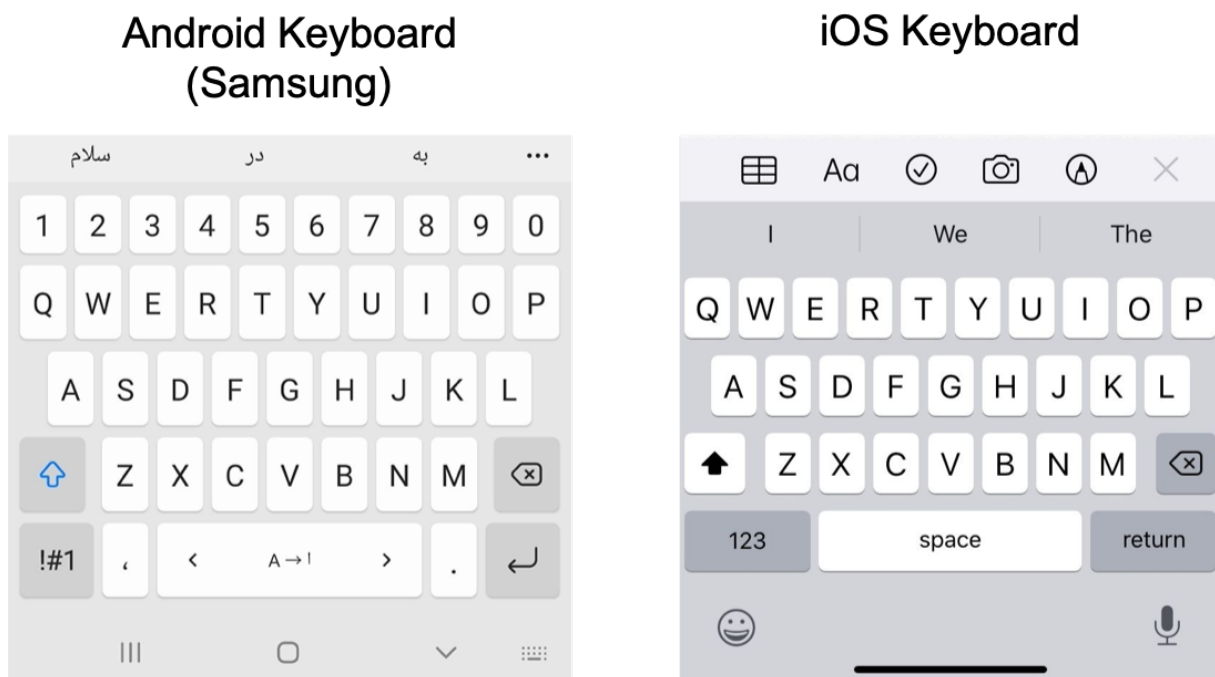
4.5 Tools and Technology Stack

Modern web applications vary in size, architecture, and technology stack, but they all share three core technologies essential for rendering a webpage in a browser, as

stated by Shahzad (2017). The first and most fundamental is HyperText Markup Language (HTML), which defines the page's elements and structure, often referred to as the Document Object Model (DOM). The second is Cascading Style Sheets (CSS), responsible for the styling, decoration, and animation of DOM elements. Lastly, JavaScript ensures the functionality and communication of the web application. While CSS and JavaScript are not strictly required for a browser to render a webpage, applications made solely with HTML are outdated and do not suit modern user expectations.

Since the research software assigned to present participants with a virtual shuffled keyboard, which is naturally an interactive component, all three technologies—HTML, CSS, and JavaScript—must be integrated into the resulting web application.

Figure 9. Default on-screen keyboards of iOS and Android platforms



As shown in Figure 9, the default virtual keyboard on both iOS and Android smartphones has 28 similar buttons representing each letter of the Latin alphabet, capable of displaying either lowercase or uppercase letters depending on the state of the case-control button. Additionally, there are mode control buttons that switch the input from letters to numbers and symbols. The need to replicate this repetitive, state-based nature of the virtual keyboard for the research set-up prompted the decision to

use a component-based web development framework, rather than the standard HTML-CSS-JavaScript bundle.

Ollila et al (2022) explored several popular and well-supported frameworks, including Angular, React, Vue, Svelte, and Blazor. Their performance comparison revealed that React outperformed the others in most scenarios involving the rendering of dynamic content. Additionally, React is the world's most preferred open-source web development framework, offering the largest knowledge base for developers to troubleshoot and solve issues. Thus, React was selected as the framework for developing the front-end part of the project.

Similar to the JavaScript frameworks discussed earlier, CSS frameworks are designed to simplify and accelerate the development of web application styles by providing concise class components. Al Salmi (2023) analysed and compared the most popular CSS frameworks from two categories: high-level, component-based frameworks like Bootstrap, Foundation, and Bulma, which offer fully styled interface elements such as buttons, dropdowns, and modals; and utility-first, low-level frameworks like Tailwind and Skeleton, which provide small, single-purpose utility classes for direct element styling. Since reproducing a virtual keyboard with specific styles and behaviours is not a trivial task, the component-based frameworks were considered unsuitable as they lack the necessary button variations. Therefore, more customisable utility-first frameworks were chosen for this project. Al Salmi (2023) found that Skeleton, being lightweight and minimalistic, offers a limited set of styling classes and is not ideal for all-purpose styling. Consequently, the more powerful open-source framework, Tailwind, was selected as the CSS framework for the web application.

Given the resource and time constraints for completing the study, the project's technology stack was kept as minimal as possible. Therefore, JavaScript was selected to build both the client and server parts of the web application. Although several runtime environments can execute JavaScript on the server side, Node.js was chosen due to its popularity and extensive package ecosystem, with over a million modules available. Nkenyereye and Jang (2016) compared Node.js's performance against the earlier Java-based JavaScript interpreter, Rhino, on key metrics such as request throughput and average response time. Eventually, they proved Node.js's superior performance in both areas. When compared to one of the newer platforms, Bun, Node.js still delivers consistently high runtime performance, despite Bun's faster startup time. Considering Node.js's reliability, scalability, high throughput, vast

ecosystem, and well-established community, this technology was selected as the server environment for this project.

Since the primary goal of the research facility is to collect primary data, selecting an appropriate database (DB) technology was an important decision. Given that the web application requires a one-sided Server-DB data flow, the key criteria for choosing the DB were schema flexibility and compatibility with the overall project technology stack. The two main types of databases considered were relational (SQL) and NoSQL. SQL databases are strict in terms of schema design and attribute typisation, making dynamic changes nearly impossible, and they also require a time-consuming normalization process, which is not necessary for noSQL databases. Regarding compatibility with the web application stack, SQL databases require data to be transformed into query formats, whereas some NoSQL databases, like MongoDB, handle data in JSON format, allowing direct manipulation without transformation from the client side. Additionally, MongoDB being a part of the widely used MERN stack for web development, makes it a suitable choice for this study's project.

The code for the web application in this study was written using Visual Studio Code (VSCode), a widely used Integrated Development Environment (IDE) by professional developers, as highlighted by a comprehensive survey by Stack Overflow (2023). This free, open-source code editor supports major programming languages and enhances the development process with features like an integrated terminal, built-in debugging environment, and an extensive extension marketplace.

4.6 UI & UX Design

The fast pace of modern life, combined with the high competition for users' attention, makes User Interface (UI) and User Experience (UX) design critical factors that ensure an application successfully serves its purpose. Research by An and Meenan (2016) driven by Google demonstrates the strong correlation between UI/UX and key product metrics. For example, they found that cluttered web pages with too many elements demonstrate a drop in conversion rate by up to 95%. Similarly, an article by Delgado (2018), which focused on the impact of web forms, revealed that 27% of users abandon long forms, and 67% never return to them.

These findings are highly relevant to the web application developed in this study, as the core UI elements are password and PIN code input forms. Since users will be typing

with a non-standard visual keyboard, which could cause puzzlement on its own, it was decided to separate the password and PIN code input forms into two distinct web pages and keep them concise. The final UI designs for both pages are shown in Figure 10.

Figure 10. User interface design applied for experiment environment

The figure displays three distinct user interface panels for password and PIN creation.

- Left Panel: Create a new password**
 - Header: "Create a new password" and "Do not use your active passwords".
 - Criteria: "Your password must satisfy the criteria" with a checklist:
 - At least 8 characters
 - Both upper and lower case letters
 - At least 1 number
 - Passwords match
 - Inputs: "New Password" and "Confirm Password" fields with visibility toggles.
 - Navigation: "Prev" and "Next" buttons.
 - Keyboard: A custom shuffled keyboard with letters like '3', '4', '5', '1', '0', '7', '9', '6', '2', '8' in the top row, and 'k', 'r', 'c', 'p', 'v', 'h', 'd', 'q', 'o', 'a' in the second row.
- Middle Panel: Create a 4-digit PIN code**
 - Header: "Create a 4-digit PIN code" and "Do not use your active PIN codes".
 - Inputs: "New PIN code" and "Confirm PIN code" fields with visibility toggles.
 - Navigation: "Prev" and "Submit" buttons.
- Right Panel: Confirmation**
 - Header: "Do not use your active PIN codes".
 - Inputs: "New PIN code" and "Confirm PIN code" fields with visibility toggles.
 - Navigation: "Prev" and "Submit" buttons.
 - Keyboard: A standard numeric keypad (0-9) with a backspace button.

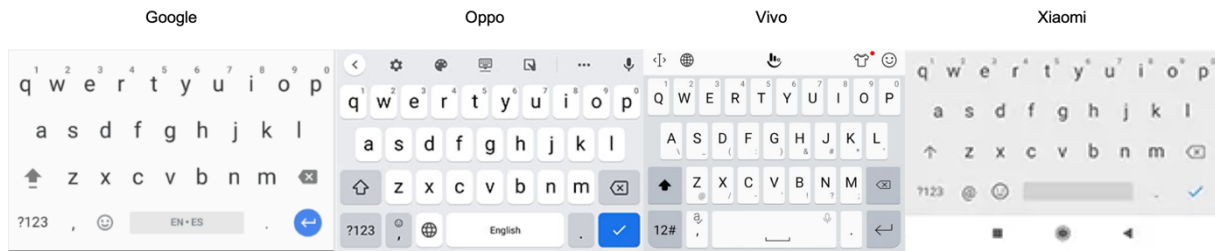
The web forms for this study were developed in strict compliance with the Web Content Accessibility Guidelines established by W3C (2023), a key requirement for meeting UK Government Accessibility Requirements. Therefore, features such as a password visibility toggle, an input erase button, and a password policy checklist were integrated into the UI design.

Additionally, the industry standard of including a password confirmation input field was applied in this project. Given the strict password policy and the potential difficulty participants might face with the novel shuffled keyboard, this measure was implemented to help users create realistic, memorable passwords while reducing the likelihood of typos.

4.7 Keyboard Design

All modern smartphones equipped with touchscreens come with at least one built-in virtual keyboard, meaning that the majority of users have extensive experience interacting with on-screen keyboards and expect them to behave in a standard way. However, a detailed examination of default keyboards installed by various vendors reveals differences in button layout and design, as illustrated in Figure 11.

Figure 11. Variation of default Android keyboard designs



Interestingly, a platform comparison shows a noticeable variation in keyboards among Android-based smartphone models, whereas iOS devices share a unified keyboard that is adjusted based on screen resolution.

Although the research software in this study is platform-agnostic and intended to collect data from users of various smartphone models, resource limitations prevent the development of more than one design for the shuffled keyboard. Given that iOS smartphones hold the largest market share, as indicated by the StatCounter (2019), and have a consistent keyboard across all devices, this keyboard was chosen as the reference for developing the shuffled keyboard in the study.

During the development of the shuffled keyboard with a design and behaviour similar to that of the iOS platform, two challenges emerged. The first challenge involved determining button sizes and the spacing between them. Since no detailed specifications are available for the default iPhone and iPad keyboards, these parameters were approximated based on elements ratio and adjusted to screen resolution.

The second challenge was selecting an appropriate font that would represent uppercase and lowercase letters, numbers, and symbols similarly. While the "San Francisco" typeface used on the iOS keyboard is available for downloading, it is a proprietary Apple product and cannot be used on devices from other vendors due to its licensing restrictions. As a result, the "Inter" font, developed by Rasmus Andersson and licensed under the Open Font License, was chosen for the shuffled keyboard design.

4.8 Software Development

Routing. When developing a web application using the React framework, selecting an appropriate routing schema to meet navigation requirements is crucial. The experiment environment in this study includes the following pages:

- Home page. The starting page of the web application, containing the participation consent form.
- Password page. A page for user-generated password input.
- PIN code page. A page for user-generated PIN code input.
- Experiment completion page. Displays a confirmation message upon completion of the experiment.
- Wrong path page. A placeholder for any URLs outside the defined path within the project domain.

The Client/Server Architecture adopted for this project often involves server-side routing through the implementation of an Application Programming Interface (API). Although server-side routing offers better performance and is favoured by search engines, as noted by Fadhilah Iskandar et al (2020), it requires serving each defined URL with its own independent frontend source bundle. This means the HTML file must be divided into several self-contained sub-documents, demanding more coding.

In contrast, React supports the MVP architectural design pattern, as previously mentioned, and allows for client-side routing, which can be managed by the Presenter component. This approach keeps the project repository more concise and executes routing logic closer to the corresponding UI elements, facilitating faster development and easier bug tracking. Given the relatively small size of the experiment software and the strict time constraints, the client-side routing approach was selected for this study.

State Management. To ensure the primary data collected within an experiment is consistent and complete React state management function was utilised. This function tracks all changes made by participants to the input data and dynamically updates the corresponding state variables. This approach enabled real-time data validation while reducing data transactions to a single POST request at the final stage of the experiment. However, a key drawback of client-side dynamic state management is its vulnerability to web application reloads. Such scenarios can occur due to user-initiated page refreshes or returning to an inactive page after a session timeout, which results in the loss of user progress. Given the experiment was designed to last no longer than

10 minutes, these scenarios were deemed low probability, and no specific workarounds were implemented to address them.

Device Detection. As mentioned earlier, this study is focused solely on smartphone users, making device detection an essential function for the experiment environment. However, there is no method that guarantees 100% accuracy in distinguishing between mobile devices and desktop/laptop computers. Cassel et al (2021) compiled and structured various available methods for web device detection, including User-Agent configuration, screen resolution, available motion and gesture events, and screen orientation. A combination of these approaches has been used to create algorithms with a satisfactory accuracy rate for identifying mobile devices.

However, the W3.org (2015) later identified that these methods were being misused for unsanctioned user tracking, a malicious practice known as browser fingerprinting. In response to these privacy concerns, Google initiated the User-Agent reduction program, which has led to a decrease in device detection accuracy. Therefore, although the developed software in this study attempts to detect mobile devices there is an acknowledged possibility that the web application could be mistakenly rendered on a desktop or laptop computer.

Default keyboard prevention. As mentioned in the previous section, the shuffled virtual keyboard based on the novel algorithm developed in this study required reconstructing the iOS keyboard, as the default keyboard cannot be altered through a web application. This essentially means that the custom keyboard is automatically displayed instead of the native one when interacting with the web application's UI elements on any mobile device. To achieve this behaviour, two approaches were considered.

The first approach involved using the *preventDefault()* method of the JavaScript Event object, which cancels the default actions triggered when an `<input>` element is focused. However, further investigation revealed inconsistent and unstable keyboard behaviour with this method, raising discussion among developers. As a result, the alternative approach of applying the "readonly" attribute to the `<input>` HTML element, which prevents the keyboard from automatically popping up, was chosen. While this method was effective, it required expanding the code base to handle events such as focusing, unfocusing, text input, and deletion explicitly.

Button Implementation. The implementation of the keyboard buttons raised several challenges. First, the empirical study by Jung and Im (2015) showed that due to the limited visual perception of small UI elements, their touch area needs to be larger than that of medium-sized elements. To reduce the probability of input errors due to touch inaccuracy while maintaining clear visual separation between buttons, as seen in the reference iOS keyboard, a transparent but clickable outline was added to the input buttons CSS code.

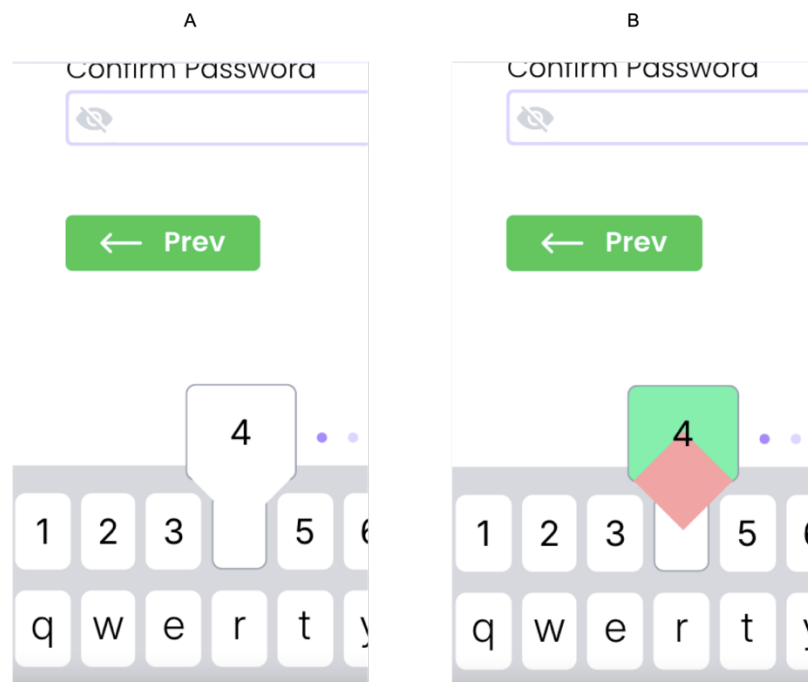
The next challenge involved replicating the complex shape of the active button on the reference keyboard, as shown in Figure 12.

Figure 12. Design of activated iOS keyboard button



During the development phase, several standard methods for creating complex shapes in HTML elements were considered. One option was the "clip-path" CSS property, which allows transforming a basic shape into a more complex one through parameterisation. However, since the required shape is essentially a polygonal model with unknown parameters, and given the lack of expertise in creating polygons, this method was dismissed as too time-consuming. Another approach was to use the `<svg>` container element, which can create a flexible viewport with custom coordinates. However, `<button>` element is not supported within `<svg>` containers, so this method was also ruled out. Ultimately, a simplified approach was used, layering multiple HTML elements with adjusted styles, positioning, and transformations to recreate the active button shape. The detailed construction of the figure is shown in Figure 13(B). While the outcome is not an exact match to the reference shape, the general design pattern was maintained.

Figure 13. Design of activated shuffled keyboard button: A – general view; B – implementation details



The final challenge during the keyboard development phase involved button animation. By default, `<button>` element switches to an active state between the *touchStart* and *touchEnd* events. However, Antal and Buza (2023) indicated that the average keypress duration is 76.89 ms, which is too short for the user to notice the button animation or receive visual confirmation of the action. To address this issue, an intentional keypress delay of an additional 150 ms was introduced to allow for confident typing.

Single Visit Control. During the experiment phase of this study, participants interacted with the developed web application to facilitate primary data collection. To ensure the representativeness of the sample, it was necessary to maintain data heterogeneity by source. Therefore, a non-personal identifier was required to differentiate new participants from those who had already taken part. This is typically achieved by storing small chunks of data in one of the browser's designated storage systems.

The most common method involves storing data in Cookies, which offers flexibility and allows for effective client state management across sessions. However, Cookies require data duplication on the server side for client matching, which reduces user privacy and anonymity. Additionally, Cookie usage is subject to strict government regulations.

An alternative, Session Storage, does not require the web application to retain data after the session ends, which is ideal for privacy. However, since session storage only keeps data within the active browser session, it is unsuitable for this study.

Thus, Local Storage was chosen as the appropriate solution. It allows data to persist until manually deleted, making it ideal for storing a unique identifier for clients who successfully completed the experiment.

However, due to the browser-dependent nature of this approach, there is a possibility of repeated participation if a user accesses the web application through a different browser or device.

4.9 Software Deployment

Once both the client (frontend) and server (backend) components of a web application have been developed and successfully tested, the next step is deployment design. Given the small size of the software in this study and the low anticipated workload, only monolithic deployment architectures such as Single Process Monolith, Modular Monolith, and Distributed Monolith were considered. According to Newman (2019), these architectures offer advantages such as simplicity in development, easier security maintenance, and better system performance. However, they are often vulnerable to challenges related to coupling and cohesion—meaning that tightly interconnected components increase the risk of errors during updates. Since the business logic of the developed web application only involves two transactions between the client and server—serving the React application on the initial GET request and receiving participant data via a POST request at the experiment’s completion—the risk of update failures was considered minimal. Therefore, the Single Process Monolith deployment design was selected for this project.

Once deployed, most software continues to evolve with product improvements, bug fixes, and code refactoring. During the development and deployment phases of this study, the web application underwent several updates as well. A Version Control System (VCS) is commonly used to track and manage changes in the code repository. Zolkifli et al (2018) identified two types of VCS: early centralised systems like Subversion and modern distributed systems like Git, noting the undeniable advantages of Git in collaborative work, branching, merging, and open-source development. A comparative analysis by Deepa et al (2020) concluded that Git was the best available option. Along with the widely used online platform GitHub, which offers a graphical

interface, Git technology was used for version control in this project. The source code is publicly available at the following link [<https://github.com/Veronika-Chupova/research-project-MERN>].

The final step in the deployment process involved selecting a suitable hosting platform for running the web application and ensuring its continuous online availability. Among the numerous cloud hosting providers, only a few offer simplified setups specifically designed for projects built with the MERN stack (MongoDB, Express, React, Node.js), which was used in this study. This allowed narrowing the options down to five platforms: Cloudways, Hostinger, Heroku, Netlify, and Vercel. After evaluating ease of setup, Heroku was chosen as the hosting platform for this project, as it provides nearly 8,000 build packs for one-click web server installation, making it ideal for deploying this application.

The publicly accessible deployment of the developed web application with the implemented shuffled keyboard is available at the link [<https://research-project-main-7b7d74af6ebc.herokuapp.com/>].

4.10 Data Collection

As discussed in previous sections, the primary data collected during the experiment phase is stored in the NoSQL MongoDB database. MongoDB is a document-oriented database, meaning that information related to each object is typically stored in a single record, known as a document.

To maintain user anonymity, the data collected is concise in structure yet sufficient and substantial for the research objectives. Each document in the database corresponds to a single participant and contains the following information:

- Participant device. General details about the device model, operating system, and browser version.
- Participant consent. The timestamp when the user provided participation consent.
- Participant content. The plain text password and PIN code entered by the user.
- Keyboard variant. The shuffled sequence of letters and numbers displayed on the virtual keyboard.
- Numboard variant. The shuffled sequence of numbers rendered on the numeric keypad.
- Event logs. Timestamps of user actions and navigation on the web page.

The data structure implementation is shown in Figure 14.

Figure 14. Schema of the collected data

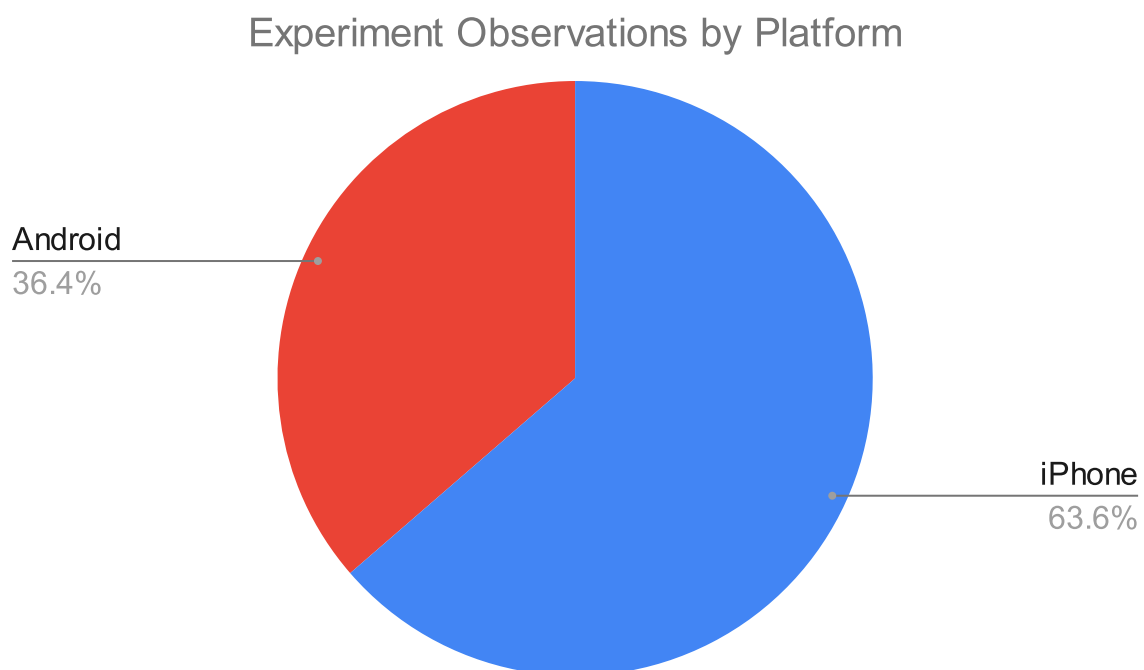
```
const sampleSchema = new mongoose.Schema ({
  device: String,
  consent: Object,
  userContent: Object,
  keyboard: Array,
  numboard: Array,
  log: Array
})
```

Chapter 5. Results Evaluation

5.1 Data Statistical Analysis

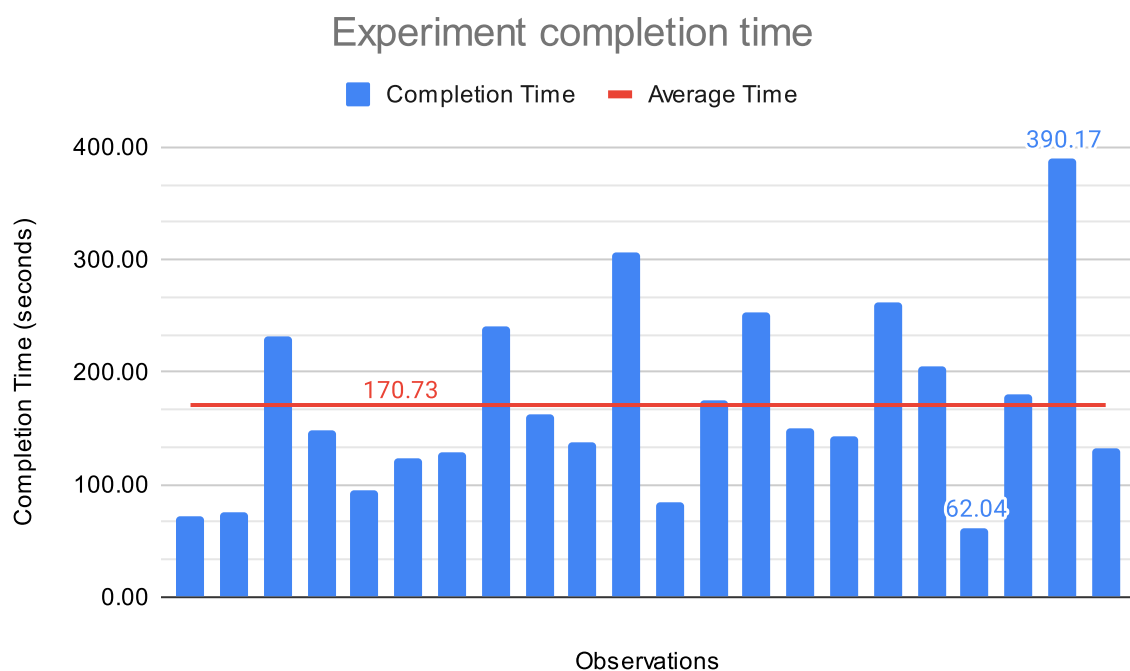
Within the experiment phase of the study, primary data was collected from a total of 22 participants. An initial analysis was conducted to verify that each entry contained a complete dataset as outlined in the Data Collection section of this report. As expected, the web application was accessed from only two mobile platforms: Android and iOS, as shown in Figure 15. Although the UserAgent method did not provide specific model information for Android devices, iOS devices were clearly identified as iPhone smartphones.

Figure 15. Source platform analysis



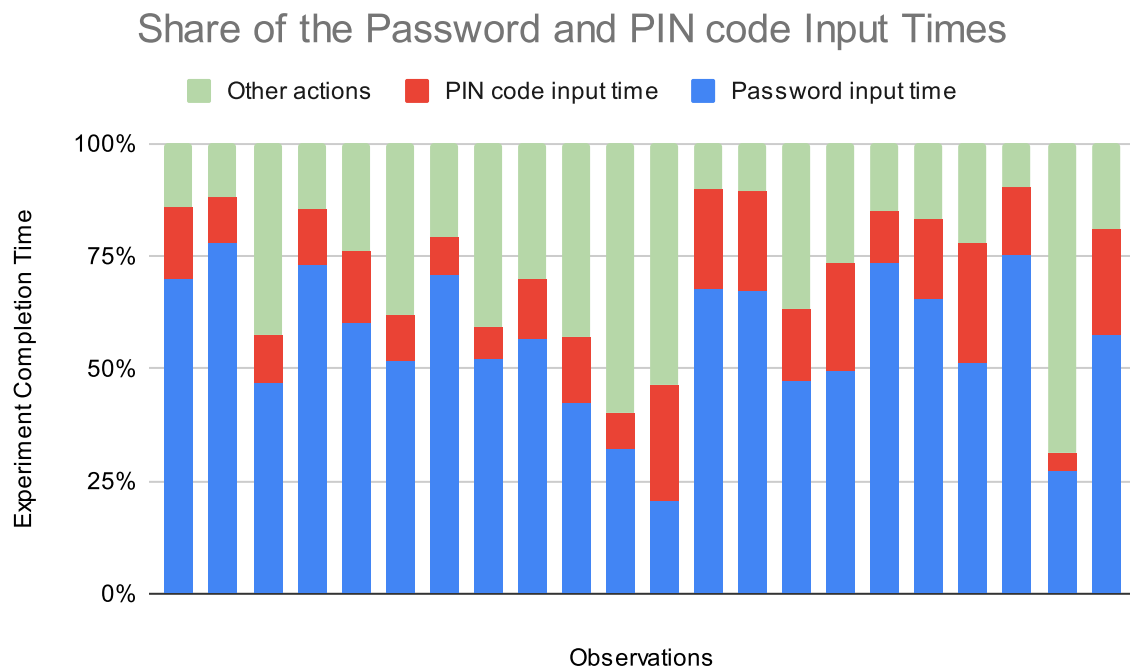
The timestamps of participants' navigation actions within the experiment platform were analysed to determine the time each user spent on every page of the web application. As shown in Figure 16, the time participants took to complete the experiment varied widely, with an average of 170.73 seconds (s) and a standard deviation of 82.81 s. The longest time a participant spent on the web application was 390.17 s or approximately 6.5 minutes, while the shortest time was 62.04 s, roughly 1 minute.

Figure 16. Experiment completion time analysis



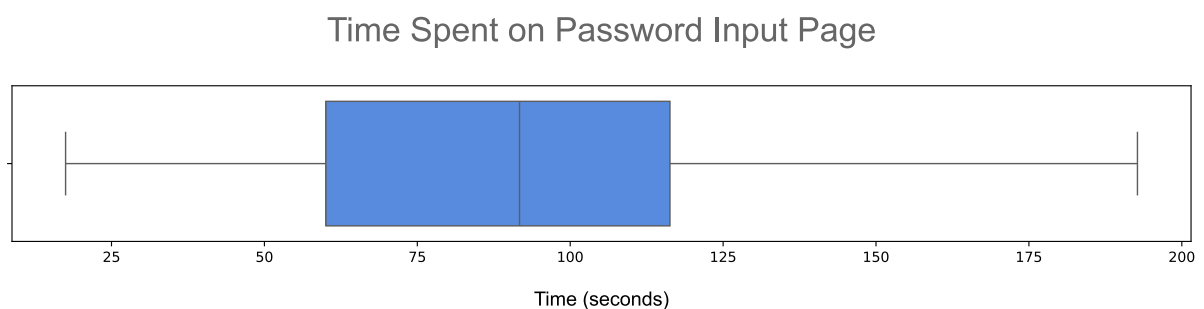
A more detailed analysis of page statistics reveals that most participants spent around half or more of their total experiment time on the password input page, significantly longer than the time spent on the PIN code input page. This likely indicates that interacting with the concise numpad was much easier for participants compared to using the full-scale shuffled keyboard. The data is illustrated in Figure 17.

Figure 17. Experiment actions time analysis



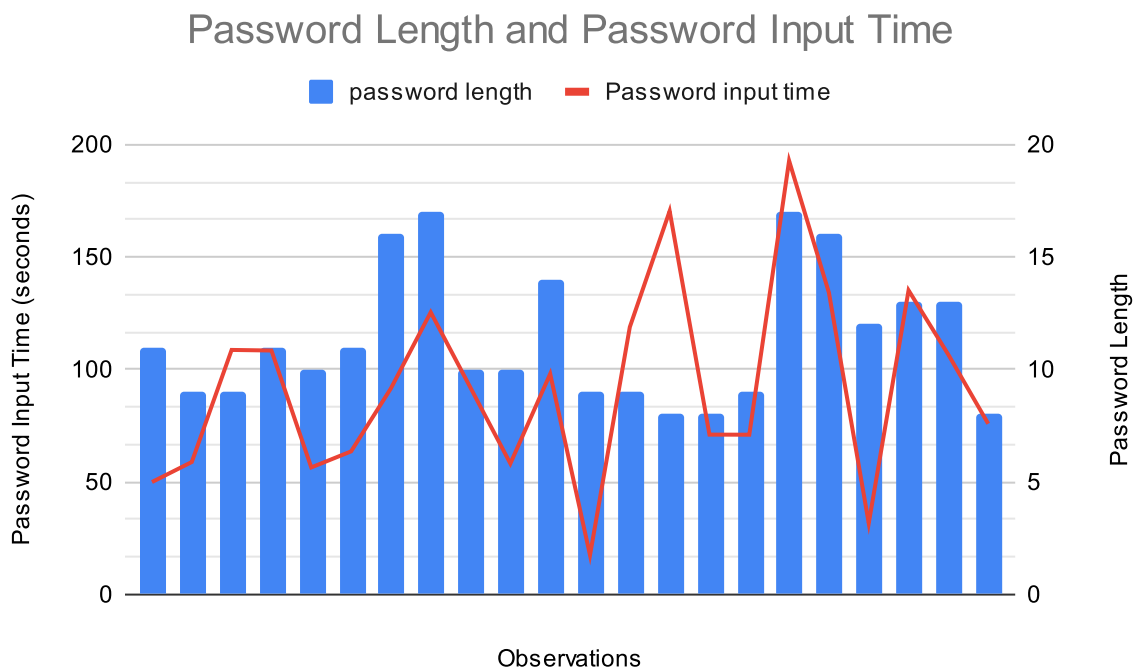
Earlier, Liu et al (2013) evaluated the time it took for individuals to enter passwords on a smartphone touchscreen using the classic QWERTY keyboard, with times ranging from 7 to 117 seconds and half of the observations (interquartile range) falling between 14 and 23 seconds. However, the data from the current experiment shows a significant shift in the interquartile range compared to the authors' findings, with the first quartile (Q1) at 60.056 seconds and the third quartile (Q3) at 116.311 seconds, as illustrated in Figure 18. Notably, the upper bound of the interquartile range in this study is very close to the maximum time participants needed to input a password using the standard QWERTY keyboard in the prior research. This suggests that participants encountered confusion and difficulty when interacting with the unfamiliar shuffled keyboard.

Figure 18. Password input time dispersion



On the other hand, the analysis of the relationship between password length and input time, shown in Figure 19, suggests that the longer time taken to enter passwords in this experiment may not be solely due to the extra time spent searching for characters on the shuffled keyboard. In fact, the correlation between these two variables is 0.44, which is considered the lower bound of a moderate correlation. However, the primary data collected in the experiment does not provide enough detail to explore this issue further.

Figure 19. Password input time and password length relation



Finally, an analysis of the composition of the collected passwords and their compliance with the study's password policy was conducted. Since the password requirements were strictly enforced—meaning participants could not proceed without providing a valid password—all strings met the criteria of being at least 8 characters long and containing at least 1 digit, 1 uppercase letter, and 1 lowercase letter. However, 19 of the passwords (86%) exceeded the minimum length requirement, with 4 of them including optional special symbols. This suggests that most participants likely aimed to create meaningful, secure passwords rather than simply entering random strings to proceed.

5.2 Passwords Sequential Pattern Analysis

This section focuses on the final objective of the research and assesses its accomplishment. To this end, the passwords and PIN codes collected during the

experiment were analysed for character sequences (patterns) derived from the standard QWERTY keyboard layout and the classic grid numpad, as shown in Figure 20.

Figure 20. Standard layouts of QWERTY keyboard and numpad



A specialised function was developed to analyse passwords for frequent patterns derived from the standard keyboard layout. This method was built on the algorithm used to validate the quality of keyboard shuffling, discussed in the earlier section of this report "Shuffling Algorithm." Although the sliding window technique was again employed, in this case, the window size varied gradually from the length of an entire keyboard row down to three characters. Additionally, the function matched patterns by moving across each line of the QWERTY keyboard individually.

The analysis identified two instances where standard keyboard patterns were present in passwords entered using the shuffled keyboard:

- The forward pattern "ert"
- The reversed pattern "987"

However, a closer examination of these two patterned passwords revealed that the "ert" sequence was part of a dictionary word, and "987" was part of the number 1987, likely referencing a year. Therefore, it can be concluded that no intentional QWERTY keyboard patterns were incorporated into the passwords created with the experimental shuffled keyboard.

An additional analysis was conducted on the collected data to identify any positional patterns specific to the shuffled keyboard variant used for entering each password. A function was developed for this assessment, utilising the same sliding window algorithm shown in Figure 6. The only difference was that the keyboard lines analysed were taken from the shuffled keyboard variant, which was unique to each participant

in the experiment. This analysis uncovered four instances where shuffled keyboard patterns were incorporated into passwords:

- Forward pattern "Oetjrlp"
- Forward pattern "048716"
- Forward pattern "Nkvpoy"
- Reverse pattern "168"

None of these patterns appears to carry any meaningful connotation and was likely included in the corresponding passwords as a result of habitual behaviour to input sequentially positioned characters, as discussed earlier in this report. Additionally, the sequences '048716' and 'Nkvpoy' were part of the same password, clearly indicating the participant's intention to use this approach.

For each password containing sequential strings derived from the shuffled keyboard, a corresponding set of characters from the standard QWERTY keyboard was generated by maintaining the same positional order of the characters. The visual representation of this translation method is shown in Figure 21.

Figure 21. Translation algorithm for keyboard positional sequential patterns

Translation Algorithm for Keyboards Sequential Patterns

Sequential Pattern of Shuffled keyboard 'Jcykl4017'



Sequential Pattern of QWERTY keyboard 'Zxcvb1234'



To evaluate the security of passwords containing sequential patterns from the shuffled keyboard, three widely used Password Strength Meters (PSM) were utilised: NIST entropy, Bitwarden, and the zxcvbn test. The latter, proposed by Wheeler and Lowe (2016) and later adopted by Dropbox, is regarded as the most reliable, according to Thai and Tanaka (2024). The translated strings, where the sequential patterns from the

shuffled keyboard were mapped onto QWERTY keyboard sequences, were also assessed using these PSMs to identify any differences in their robustness. Table 2 presents the strength evaluation results for each password containing a sequential pattern. Though, each employed PSM applies own measurement system, the overall principle doesn't change—the more points gained, the stronger the string. Additionally, the zxcvbn test provides roughly estimated time

Table 2. Sequential strings' strength comparison

		NIST Entropy	Bitwarden	zxcvbn test
Password #1 with pattern 'Oetjrlp'	Original string	31.5	good	3/4 (centuries)
	QWERTY representation	25.5	very weak	1/4 (6 days)
Password #2 with patterns '048716' and 'Nkvpoy'	Original string	36	strong	4/4 (centuries)
	QWERTY representation	30	very weak	1/4 (4 days)
Password #3 with pattern with '168'	Original string	31.5	good	3/4 (centuries)
	QWERTY representation	31.5	weak	2/4 (8 years)

Evidently, the poor practice of incorporating positional keyboard sequences leads to the creation of strong, secure passwords when using the shuffled keyboard proposed in this study. However, the same sequences on a standard QWERTY keyboard result in weak passwords that are vulnerable to both brute force and dictionary attacks.

The remaining 19 passwords that don't contain positional keyboard patterns were also assessed with the aforementioned meters and results are placed in Table 3. While about 70% of them are considered sufficiently robust (good and higher in Bitwarden, 3 and 4 in the zxcvbn test), 30% were marked as not secure. It is worth mentioning that the majority of these passwords include dictionary words and, thus, may appear in existing password dictionaries which explains the reached marks. However, the genuine reason for each mark is left unexplored in this study because it falls out of the research goal.

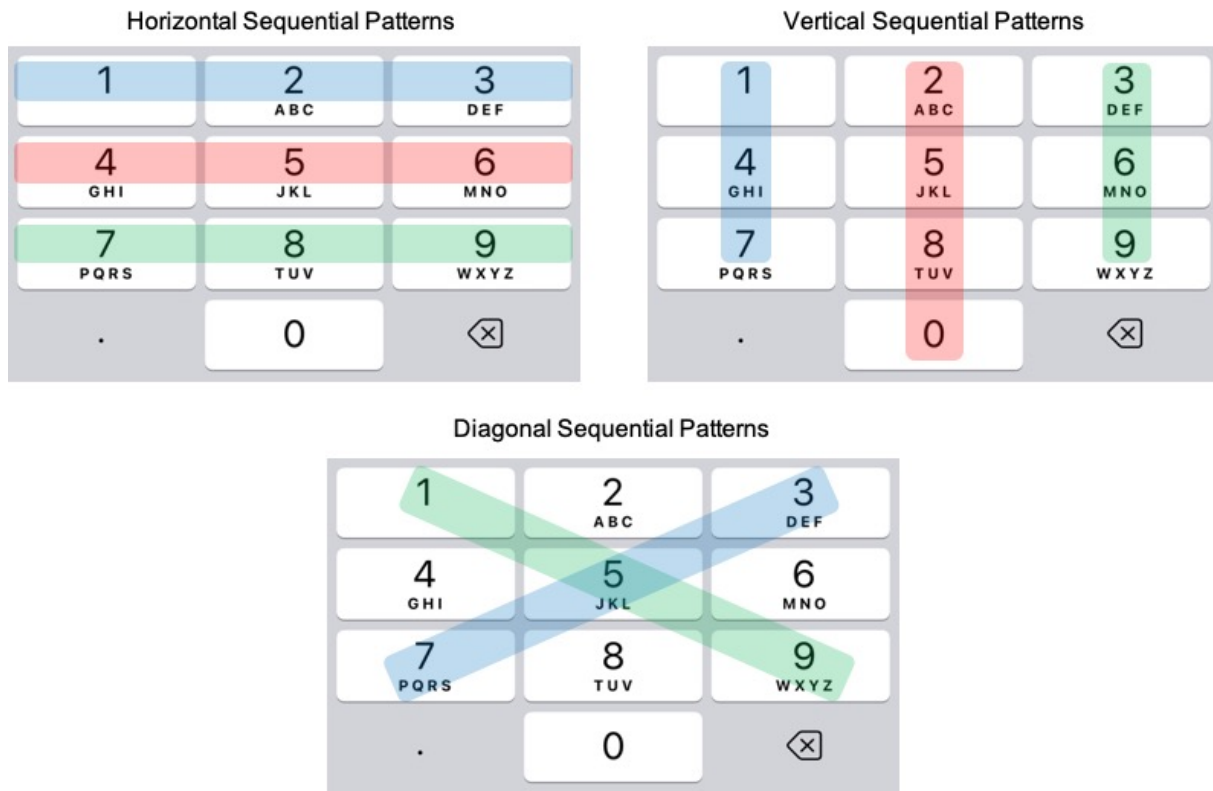
Table 3. Passwords strength

	NIST Entropy	Bitwarden	zxcvbn test
Password #1	28.5	good	3/4 (centuries)
Password #2	25.5	good	3/4 (centuries)
Password #3	28.5	strong	4/4 (centuries)
Password #4	27	weak	2/4 (7 years)
Password #5	28.5	weak	2/4 (5 years)
Password #6	36	strong	4/4 (centuries)
Password #7	43.5	strong	4/4 (centuries)
Password #8	33	good	3/4 (centuries)
Password #9	27	very weak	1/4 (1 month)
Password #10	33	strong	4/4 (centuries)
Password #11	25.5	good	3/4 (centuries)
Password #12	24	weak	2/4 (centuries)
Password #13	24	weak	2/4 (centuries)
Password #14	25.5	good	3/4 (centuries)
Password #15	43.5	strong	4/4 (centuries)
Password #16	36	strong	4/4 (centuries)
Password #17	31.5	strong	4/4 (centuries)
Password #18	37.5	strong	4/4 (centuries)
Password #19	24	weak	2/4 (centuries)

5.3 PIN codes Sequential Pattern Analysis

The common sequential patterns on the numpad follow the same principles as those on the keyboard, previously discussed in this work. The key differences lie in the shorter sequence lengths, due to the more compact layout of the standard numpad's rows, and the widespread usage of additional patterns created by vertical and diagonal sequences, as shown in Figure 22.

Figure 22. Numboard positional sequential patterns visualisation



A specific function was developed to identify numpad patterns in the collected 4-digit PIN codes. Unlike the function used for password analysis, which included extracting keyboard sequences, this function focused solely on accelerating the search for patterns, as the number of possible sequences, both forward and reversed, is relatively small.

Surprisingly, not a single sequence derived from the standard numpad was found in the collected PIN codes entered using the shuffled numpad during the experiment. This suggests that the natural numerical order and familiar sequences from the standard numpad did not influence participants' choices when selecting their PIN codes.

To investigate whether the provided 4-digit codes were influenced by positional patterns, as illustrated in Figure 22, each entry was analysed using a similar function. This time, the search focused on positional pattern sequences, both forward and reversed, derived from the unique shuffled numpad variant presented to each participant.

The analysis revealed only one instance where a vertical pattern, represented by the string '410', was incorporated into a PIN code. However, the private research by

Datagenetics.com (2012), showed that this sequence is not included in the top 20 most frequently used PIN codes, which account for 26.83% of the 3.4 million samples analysed by the author.

These findings suggest that using the shuffled numpad for PIN code entry effectively prevents the inclusion of easy-to-guess, frequent patterns from the standard numpad layout, thereby strengthening PIN codes against dictionary attacks.

Chapter 6. Conclusion

This study offers an in-depth exploration of how keyboard layout affects the strength of user-generated passwords and PIN codes, using shuffling techniques to enhance their security.

At the beginning, the importance of password strength for comprehensive data protection was emphasised, and the vulnerabilities of traditional security methods were identified. The study aimed to use shuffled keyboard layouts to eliminate weak password and PIN combinations, ultimately improving the strength of user-generated strings.

To accomplish this, a novel algorithm for keyboard shuffling was developed. The method incorporates the well-known Sattolo algorithm, along with a specifically designed entropy assessment algorithm, to shuffle the character sequences of the standard keyboard and numpad, generating sufficiently randomised layouts.

The experiment results showed that using this shuffled keyboard for password entry effectively eliminated the frequent sequences typically associated with the standard QWERTY layout. Additionally, the shuffled keyboard produced strong passwords, even when sequential patterns were deliberately included. Similarly, analysis of the PIN codes entered on the shuffled numpad revealed no standard numpad patterns or naturally ordered sequences, further strengthening the security of the generated PINs.

Throughout the study, a critical evaluation of existing methods, tools, and solutions drove key decisions and directed the design, development, and assessment of the experiment.

The study was conducted with strict adherence to academic integrity, making efficient use of the available time and resources. The work fully complied with ethical standards and legal requirements, ensuring data security and privacy while avoiding

discrimination or bias in both the algorithm development and data collection processes. Social implications were also taken into account, with a focus on promoting accessibility for all users.

6.1 Research Limitations

While the research aim is considered achieved, several limitations must be acknowledged. First, the dataset collected during the limited experiment time includes data from 22 participants, which may limit the generalisability of the findings to a wider population. Second, the sampling technique primarily relied on snowball recruitment, where existing participants invited others from their social networks. Since early participants were drawn from individuals close to the author, this approach may introduce selection bias by potentially excluding individuals from more distant social classes and backgrounds.

Due to the strict time constraints of the experiment, the long-term memorability of the collected passwords and PIN codes, reflecting the practical applicability of these secret strings, may have been overlooked. Another limitation is the English-centric design of the study. Using only one language may narrow the insights, as different languages and writing systems can significantly impact password creation patterns and user behaviour.

Additionally, resource limitations restricted the integration of a more advanced web tracking system, which would have enabled the collection, storage, and analysis of detailed user activity data across the web application, potentially offering deeper insights into behaviour and data patterns.

Finally, ethical considerations prevented the direct use of numerous leaked password and PIN code databases, so the experiment relied on summarised findings from previous research. As a result, some keyboard-derived patterns that affect password strength may have been missed, making the study's findings less precise.

6.2 Future Directions

While this study offers valuable insights into the effectiveness of the shuffled keyboard in generating user-created passwords and PIN codes, several areas require further research. Future studies should include a larger and more diverse sample size to improve the generalisability of the findings across different populations. Additionally,

incorporating language variation into the study design could help assess the universality of the shuffled keyboard approach and identify potential challenges related to different languages, particularly those using logographic (e.g., Japanese) or abjad (e.g., Arabic) writing systems.

Extending the study's duration would also allow for an exploration of the long-term memorability of passwords and PIN codes entered using the shuffled keyboard within a broader experimental framework. Finally, integrating a detailed action tracking system could provide deeper insights into user-keyboard interactions and ultimately improve the overall user experience.

References

- Agarwal, A.K., Rani, L., Tiwari, R.G., Sharma, T., Sarangi, P.K. (2021). Honey Encryption: Fortification Beyond the Brute-Force Impediment. In: Manik, G., Kalia, S., Sahoo, S.K., Sharma, T.K., Verma, O.P. (eds) *Advances in Mechanical Engineering. Lecture Notes in Mechanical Engineering*. Springer, Singapore.
https://doi.org/10.1007/978-981-16-0942-8_64
- Agarwal, M., Mehra, M., Pawar, R. and Shah, D. (2011). Secure authentication using dynamic virtual keyboard layout. *Proceedings of the International Conference & Workshop on Emerging Trends in Technology - ICWET '11*.
doi:<https://doi.org/10.1145/1980022.1980087>.
- Al Salmi, H. (2023). Comparative CSS frameworks. [online] *Multi-Knowledge Electronic Comprehensive Journal For Education And Science Publications*. Available at: https://mecsj.com/uplode/images/photo/hat4_.pdf.
- Alkhwaja, I., Albugami, M., Alkhwaja, A., Alghamdi, M., Abahussain, H., Alfawaz, F., Almurayh, A., and Min-Allah, N. (2023). Password Cracking with Brute Force Algorithm and Dictionary Attack Using Parallel Programming. *Applied Sciences*, 13(10), pp.5979–5979. doi:<https://doi.org/10.3390/app13105979>.
- An, D. and Meenan, P., 2016. Why marketers should care about mobile page speed. [online] Available at: https://www.thinkwithgoogle.com/qs/documents/2698/53dff_Why-Marketers-Should-Care-About-Mobile-Page-Speed-EN.pdf [Accessed 24 Aug. 2024]
- Antal, M. and Buza, K., 2023. SapiPin: Observations on PIN-code typing dynamics. *Acta Universitatis Sapientiae, Informatica*, 15(1), pp.10-24.
- Aylmer, R. (1938). *Statistical Tables for Biological, Agricultural and Medical Research*, By R.A. Fisher and F. Yates.
- Brindha Devi, V., Sindhuja, S., Shanthini, S. and Hemalatha, M. (2018). A virtual keyboard security system for automated teller machine. *International Journal of Engineering & Technology*, 7(3.3), p.59.
doi:<https://doi.org/10.14419/ijet.v7i2.33.13855>.
- Cassel, D., Lin, S.-C., Buraggina, A., Wang, W., Zhang, A., Bauer, L., Hsiao, H.-C., Jia, L. and Libert, T. (2021). OmniCrawl: Comprehensive Measurement of Web Tracking With Real Desktop and Mobile Browsers. *Proceedings on Privacy Enhancing Technologies*, 2022(1), pp.227–252. doi:<https://doi.org/10.2478/popets-2022-0012>.

Carnavalet, X.D.C.D. and Mannan, M. (2015). A Large-Scale Evaluation of High-Impact Password Strength Meters. *ACM Transactions on Information and System Security*, 18(1), pp.1–32. doi:<https://doi.org/10.1145/2739044>.

Chromium.org. (2023). User-Agent Reduction. [online] Available at: <https://www.chromium.org/updates/ua-reduction/> [Accessed 12 Jul. 2024].

Chou, H.-C., Lee, H.-C., Hsueh, C.-W. and Lai, F.-P. (2012). Password Cracking Based on Special Keyboard Patterns. [online] Available at: <http://tkuir.lib.tku.edu.tw:8080/dspace/bitstream/987654321/80105/2/PasswordCrackingBasedonSpecialKeyboardPatterns.pdf> [Accessed 28 Jun. 2024].

Consumer Rights Act (2015). [online] Available at: <https://www.legislation.gov.uk/ukpga/2015/15/contents> [Accessed 27 Jun. 2024]
Datagenetics.com (2012). PIN number analysis. [online] Available at: <http://www.datagenetics.com/blog/september32012/>.

Data protection Act (2018). [online] Available at: <https://www.legislation.gov.uk/ukpga/2018/12/contents/enacted> [Accessed 10 Jul. 2024]

Delgado, M. (2018). 6 Steps for Avoiding Online Form Abandonment. [online] Themanifest.com. Available at: <https://themanifest.com/web-design/blog/6-steps-avoid-online-form-abandonment> [Accessed 17 Sep. 2024].

Durstenfeld, R. (1964). Algorithm 235: Random permutation. *Communications of the ACM*, 7(7), p.420. doi:<https://doi.org/10.1145/364520.364540>.

Fadhilah Iskandar, T., Lubis, M., Fabrianti Kusumasari, T. and Ridho Lubis, A. (2020). Comparison between client-side and server-side rendering in the web development. *IOP Conference Series: Materials Science and Engineering*, 801, p.012136. doi:<https://doi.org/10.1088/1757-899x/801/1/012136>.

Forbes (2022). Thanks To Apple, Microsoft And Google Passwords Will Finally Die. [online] Available at: <https://www.forbes.com/sites/davidbirch/2022/07/25/thanks-to-apple-microsoft-and-google-passwords-will-finally-die/> [Accessed 19 Sep. 2024]

GDPR (2018). General data protection regulation (GDPR). [online] General Data Protection Regulation (GDPR). Available at: <https://gdpr-info.eu/>.

Geofrey, M., Nyabuto, M., Mony, M. and Mbugua, S. (2024). Architectural Review of Client-Server Models. *International Journal of Scientific Research & Engineering*

Trends, [online] 10(1), pp.2395–566. Available at: https://ijsret.com/wp-content/uploads/2024/01/IJSRET_V10_issue1_125.pdf.

Glory, F. Z., Ul Aftab, A., Tremblay-Savard, O., and Mohammed, N. (2019). Strong Password Generation Based On User Inputs. IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 2019, pp. 0416-0423, doi: 10.1109/IEMCON.2019.8936178.

GOV.UK (2018). Data Protection Act 2018. [online] Legislation.gov.uk. Available at: <https://www.legislation.gov.uk/ukpga/2018/12/contents/enacted>.

Grassi, P.A., Garcia, M.E. and Fenton, J.L., 2017. Digital Identity Guidelines (翻訳版). NIST special publication, 800, pp.63-3.

Güven, E.Y., Boyaci, A. and Aydin, M.A. (2022). A Novel Password Policy Focusing on Altering User Password Selection Habits: A Statistical Analysis on Breached Data. Computers & Security, 113, p.102560. doi:<https://doi.org/10.1016/j.cose.2021.102560>.
Kanta, A., Coisel, I. and Scanlon, M. (2022). A Novel Dictionary Generation Methodology for Contextual-Based Password Cracking. IEEE Access, 10, pp.59178–59188. doi:<https://doi.org/10.1109/access.2022.3179701>.

Jr, S. (2015). SPA Design and Architecture. Simon and Schuster.

Jung, E.S. and Im, Y. (2015). Touchable area: An empirical study on design approach considering perception size and touch input behavior. International Journal of Industrial Ergonomics, 49, pp.21–30. doi:<https://doi.org/10.1016/j.ergon.2015.05.008>.

Karaman, E. and Kurt, M., 2015. Comparison of project management methodologies: prince 2 versus PMBOK for it projects. Int. Journal of Applied Sciences and Engineering Research, 4(4), pp.572-579.

Koegel, J.F., Hong Heather Yu and Eng Keong Lua (2009). P2P networking and applications. Amsterdam ; Boston: Elsevier/Morgan Kaufmann.

Liu, D., Cuervo, E., Pistol, V., Scudellari, R. and Cox, L.P. (2013). ScreenPass. doi:<https://doi.org/10.1145/2462456.2465425>.

Lowe Wheeler, D. and Lowe, D. (2016). Open access to the Proceedings of the 25th USENIX Security Symposium is sponsored by USENIX zxcvbn: Low-Budget Password Strength Estimation zxcvbn: Low-Budget Password Strength Estimation. [online] Available at: https://www.usenix.org/system/files/conference/usenixsecurity16/sec16_paper_wheeler.pdf.

Lustro, R.A.F. (2019). Ameliorating Password Security Authentication Using BCrypt Algorithm with Dynamic Salt Generation. *Journal of Advanced Research in Dynamical and Control Systems*, 11(12-SPECIAL ISSUE), pp.1240–1245.
doi:<https://doi.org/10.5373/jardcs/v11sp12/20193331>.

Lyu, S., Yao, Q. and Song, J. (2022). AvoidPwd: A mnemonic password generation strategy based on keyboard transformation. *China Communications*, 19(10), pp.92–101. doi:<https://doi.org/10.23919/jcc.2022.00.027>.

N.Deepa, B.Prabadevi, L.B, K. and B.Deepa (2020). An analysis on Version Control Systems. 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE). doi:<https://doi.org/10.1109/ic-etite47903.2020.39>.

NCSC (2018). Password administration for system owners. [online] [Ncsc.gov.uk](https://www.ncsc.gov.uk). Available at: <https://www.ncsc.gov.uk/collection/passwords/updating-your-approach>.

Newman, S. (2019). *Monolith to microservices : evolutionary patterns to transform your monolith*. Sebastopol, Ca: O'reilly Media, Inc.

Ng, D., Ho, J., Hercules, C., Bravo-Lillo, C. and Schechter, S. (2022). Do Password Managers Improve Password Hygiene? *Harvard.edu*. [online] doi:<https://nrs.harvard.edu/URN-3:HUL.INSTREPOS:37374029>.

Nkenyereye, L. and Jang, J.-W. (2016). Performance Evaluation of Server-side JavaScript for Healthcare Hub Server in Remote Healthcare Monitoring System. *Procedia Computer Science*, 98, pp.382–387.
doi:<https://doi.org/10.1016/j.procs.2016.09.058>.

Ntantogian, C., Malliaros, S. and Xenakis, C. (2019). Evaluation of password hashing schemes in open source web platforms. *Computers & Security*, 84, pp.206–224.
doi:<https://doi.org/10.1016/j.cose.2019.03.011>.

Ollila, R., Mäkitalo, N. and Mikkonen, T., 2022. Modern web frameworks: A comparison of rendering performance. *Journal of Web Engineering*, 21(3), pp.789–813.

Richards, M. (2020). *Fundamentals of software architecture : an engineering approach*. O'reilly Media, Inc.

Rodriguez, J.J., Zibran, M.F. and Eishita, F.Z. (2022). Finding the Middle Ground: Measuring Passwords for Security and Memorability. [online] *IEEE Xplore*. doi:<https://doi.org/10.1109/SERA54885.2022.9806772>.

Sattolo, S. (1986). An algorithm to generate a random cyclic permutation. *Information Processing Letters*, 22(6), pp.315–317. doi:[https://doi.org/10.1016/0020-0190\(86\)90073-6](https://doi.org/10.1016/0020-0190(86)90073-6).

Schneider, D., Otte, A., Gesslein, T., Gagel, P., Kuth, B., Damalakhi, M.S., Dietz, O., Ofek, E., Pahud, M., Kristensson, P.O., Muller, J. and Grubert, J. (2019). ReconViguRation: Reconfiguring Physical Keyboards in Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics*, 25(11), pp.3190–3201. doi:<https://doi.org/10.1109/tvcg.2019.2932239>.

Senthilselvi, A., Surya, H., Preeti Charishma, P., and Raja, K. (2024). HaShuffle-Crafting Secure Passwords with a Splash of Shuffle Magic. 2024 2nd International Conference on Networking and Communications (ICNWC) 2-4 April 2024. doi:<https://doi.org/10.1109/icnwc60771.2024.10537317>.

Shahzad, F. (2017). Modern and Responsive Mobile-enabled Web Applications. *Procedia Computer Science*, 110, pp.410–415. doi:<https://doi.org/10.1016/j.procs.2017.06.105>.

Shi, R., Zhou, Y., Li, Y. and Han, W. (2021). Understanding Offline Password-Cracking Methods: A Large-Scale Empirical Study. *Security and Communication Networks*, 2021, pp.1–16. doi:<https://doi.org/10.1155/2021/5563884>.

Singh, A. and Raj, S. (2019). Securing password using dynamic password policy generator algorithm. *Journal of King Saud University - Computer and Information Sciences*, 34. doi:<https://doi.org/10.1016/j.jksuci.2019.06.006>.

Statista Daily Data. (2019). Infographic: Millions in the UK are using vulnerable passwords. [online] Available at: <https://www.statista.com/chart/17767/most-commonly-re-occurring-uk-passwords/>.

Stack Overflow. (2023). Stack Overflow Developer Survey 2023. [online] Available at: <https://survey.stackoverflow.co/2023/#most-popular-technologies-new-collab-tools>.

StatCounter (2019). Mobile Vendor Market Share Worldwide | StatCounter Global Stats. [online] StatCounter Global Stats. Available at: <https://gs.statcounter.com/vendor-market-share/mobile/worldwide>.

Steves, M.P., Chisnell, D., Sasse, A., Krol, K., Theofanos, M.F. and Wald, H. (2014). Report: Authentication Diary Study. doi:<https://doi.org/10.6028/nist.ir.7983>.

Suood Alroomi and Li, F. (2023). Measuring Website Password Creation Policies At Scale. arXiv (Cornell University). doi:<https://doi.org/10.1145/3576915.3623156>.

Thanh, L. and Tanaka, H. (2024). A statistical Markov-based password strength meter. *Internet of Things*, 25, pp.101057–101057. doi:<https://doi.org/10.1016/j.iot.2023.101057>.

Volovyk, O. and Harmash, O. (2022). Exploring Current Project Management Methodologies in the Context of their Best Applications. *Electronic Scientific Journal Intellectualization of Logistics and Supply Chain Management* #1 2020, 14(14), pp.17–30. doi:<https://doi.org/10.46783/smart-scm/2022-14-2>.

Wang, D., Zhang, Z., Wang, P., Yan, J. and Huang, X. (2016). Targeted Online Password Guessing. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security - CCS'16*. doi:<https://doi.org/10.1145/2976749.2978339>.

W3.org. (2015). Unsanctioned Web Tracking. [online] Available at: <https://www.w3.org/2001/tag/doc/unsanctioned-tracking/#unsanctioned-tracking-tracking-without-user-control> [Accessed 17 Jul. 2024].

W3C (2023). Web Content Accessibility Guidelines (WCAG) 2.1. [online] W3.org. Available at: <https://www.w3.org/TR/WCAG21/>.

Yang, K., Hu, X., Zhang, Q., Wei, J. and Liu, W. (2021). Studies of Keyboard Patterns in Passwords: Recognition, Characteristics and Strength Evolution. *Lecture notes in computer science*, pp.153–168. doi:https://doi.org/10.1007/978-3-030-86890-1_9.

Yu, F. and Huang, Y. (2015). An Overview of Study of Password Cracking. *International Conference on Computer Science and Mechanical Automation (CSMA)*, Hangzhou, China, 2015, pp. 25-29, doi: 10.1109/CSMA.2015.12.

Zolkifli, N.N., Ngah, A. and Deraman, A. (2018). Version Control System: A Review. *Procedia Computer Science*, 135, pp.408–415. doi:<https://doi.org/10.1016/j.procs.2018.08.191>.

Appendix A. Participant Consent Form - View (1/2)

> Participant Information

1

What is the aim of the study?

The study aims to validate how different screen (virtual) keyboards impact the presence of frequent repetitive patterns in user-generated passwords and PIN codes.

Why have you been invited to participate?

To fulfill the research aim and objectives, participants with various background who satisfy the following criteria were invited: capable persons over 18 years old in possession of Android/iOS - based smartphone with access to the Internet and able to read in English.

Do you have to participate?

This document was provided to help your decision on whether you would like to participate in the study or not. It is completely up to you what you will decide eventually.

How many times you can participate?

The study needs to ensure data is collected once from every participant. Because of that reason, the dedicated website at [https://research-project-main-7b7d74af6ebc.herokuapp.com/] uses cookies to detect previous participation. The cookie is represented by a timestamp of your previous data submission stored exclusively in the participant's internet browser (local storage) until

✓ Terms of Use

Start

> Participant Information

2

How many times you can participate?

The study needs to ensure data is collected once from every participant. Because of that reason, the dedicated website at [https://research-project-main-7b7d74af6ebc.herokuapp.com/] uses cookies to detect previous participation. The cookie is represented by a timestamp of your previous data submission stored exclusively in the participant's internet browser (local storage) until manually deleted. The cookie only allows detecting previous openings of the website and does not allow user identification or collected data attribution.

What actions would you be required to do?

You will need to open the website [https://research-project-main-7b7d74af6ebc.herokuapp.com/] on your smartphone using an internet browser. Then you will be asked to complete two web forms on this website. The first will require generating a new password that satisfies the described criteria and inputting it twice using the suggested screen keyboard. The second one will require creating a new 4-digit PIN code and inputting it twice using the suggested screen numeric keyboard. The anticipated time to complete both forms is no more than 5 minutes.

What risks are associated with participation in the study?

When participating in the study, you are encouraged to strictly adhere to the study recommendations and not use any of your active or previously used passwords. Also, you are strongly recommended not to set a password and PIN code generated within the study participation in your future online activity as they will be processed in plain text by the researcher. Following these warnings will eliminate the risks of sensitive data compromise.

✓ Terms of Use

Start

> Participant Information

3

What risks are associated with participation in the study?

When participating in the study, you are encouraged to strictly adhere to the study recommendations and not use any of your active or previously used passwords. Also, you are strongly recommended not to set a password and PIN code generated within the study participation in your future online activity as they will be processed in plain text by the researcher. Following these warnings will eliminate the risks of sensitive data compromise.

What benefits are associated with participation in the study?

The study participation does not provide participants with any direct benefits. However, this research elaborates on password strengthening and resistance against cyber-attacks, which may help enhance data protection mechanisms in the future.

What data about you will be collected?

Within the study, any personal data or other identifiers will NOT be collected. All data collected will be anonymous (lack of any of your unique personal attributes, e.g. name, age, location, IP-address, etc.), which means it will be impossible for anyone to identify you and attribute collected data to you. Because of that reason, the data you provide cannot be withdrawn after web forms are completed and submitted. Within the study, the following data will be collected:
User-generated information: the inputted password and PIN

✓ Terms of Use

Start

> Participant Information

4

following data will be collected:

User-generated information: the inputted password and PIN code

Technical environment information: device model (e.g. iPhone X), operating system (e.g. iOS 12.4), Internet browser version (e.g. Safari)

Statistical information: time-stamps on user-system interactions such as button clicks, checkbox ticks

How will the collected data be handled and used?

Up to the end of the data collection phase on 15 September 2024, the anonymous data will be stored securely in the cloud-based database, to which only the research author has access. In the data analysis phase, the anonymous data accrued in the previous step will be moved onto a researcher's local drive outside any network and inaccessible to a third party. Then, the results of the data analysis will be published as a part of the Master's dissertation.

The data will be stored and analysed in plain text. After the University checks are completed and the research project is marked, the data will be destroyed with no recovery option.

Informed consent

Your consent to participate in the research must be fully informed and voluntary. You have the right and opportunity to ask any questions and clarify any concerns related to your participation in the given study before providing your consent. There is a contact section at the end of this document for you to address any related inquiries.

✓ Terms of Use

Start

> Participant Information

5

Informed consent

Your consent to participate in the research must be fully informed and voluntary. You have the right and opportunity to ask any questions and clarify any concerns related to your participation in the given study before providing your consent. There is a contact section at the end of this document for you to address any related inquiries.

Can you withdraw your consent on participation and collected data?

The study design aims at collecting the data anonymously. This fact excludes the possibility of identifying participants by any means. Therefore, withdrawal after completion of the assigned study tasks is unavailable due to the impossibility of mapping the data.

How can you find out about the results of the study?

The study outcomes will be available starting from 01 October 2024. A brief presentation of the key findings will be published on the website [https://research-project-main-7b7d74af6ebc.herokuapp.com/] where the participants took part in the research and will be publicly available up to 31 December 2024.

Ethical approvals

This study is performed in accordance with Edge Hill University Research Ethics Policy available at [https://www.edgehill.ac.uk/wp-content/uploads/documents/research-ethics-policy-1-1.pdf].

✓ Terms of Use

Start

> Participant Information

6

2024. A brief presentation of the key findings will be published on the website [https://research-project-main-7b7d74af6ebc.herokuapp.com/] where the participants took part in the research and will be publicly available up to 31 December 2024.

Ethical approvals

This study is performed in accordance with Edge Hill University Research Ethics Policy available at [https://www.edgehill.ac.uk/wp-content/uploads/documents/research-ethics-policy-1-1.pdf]. The research proposal has been reviewed and then granted ethical approval by the Department of Computer Science of Edge Hill University.

Contact Details:

Researcher Name: Veronika Chupova
Researcher Email Address: 25868705@edgehill.ac.uk

☐ I have read and understood "Participant Information" and give my consent to take part in the study

☐ I am a capable person over 18 years old

✓ Terms of Use

Start

Appendix A. Participant Consent Form - Text (2/2)

What is the aim of the study?

The study aims to check how different screen keyboards impact user-generated password strength and complexity.

Why have you been invited to participate?

To fulfill the research aim and objectives, participants with various background who satisfy the following criteria were invited: capable persons over 18 years old in possession of Android/iOS - based smartphone with access to the Internet and able to read in English.

Do you have to participate?

This document was provided to help your decision on whether you would like to participate in the study or not. It is completely up to you what you will decide eventually.

How many times you can participate?

The study needs to ensure data is collected once from every participant. Because of that reason, the dedicated website at [<https://research-project-main-7b7d74af6ebc.herokuapp.com/>] uses cookies to detect previous opening. The cookie is represented by unified text data stored in the participant's internet browser (local storage) until manually deleted. The cookie only allows detecting previous openings of the website and does not allow user identification or collected data attribution.

What actions would you be required to do?

You will need to open the website on your smartphone using an internet browser. Then you will be asked to complete two web forms. The first will require generating and inputting twice of a new password that satisfies the described criteria using the suggested screen keyboard. The second one will require creating and inputting twice a new 4-digit PIN code using the suggested screen numeric keyboard. The anticipated time to complete both forms is no more than 5 minutes.

What risks are associated with participation in the study?

When participating in the study, you are encouraged to strictly adhere to the study recommendations and not use any of your active or previously used passwords. Also, you are strongly recommended not to set a password and PIN code generated within the study participation in your future online activity. Following these warnings will eliminate the risks of sensitive data compromise.

What benefits are associated with participation in the study?

The study participation does not provide participants with any direct benefits. However, this research elaborates on password strengthening and resistance against cyber-attacks, which may help enhance data protection mechanisms in the future.

What data about you will be collected?

Within the study, any personal data or other identifiers will NOT be collected. All data collected will be anonymous, which means it will be impossible for anyone to identify you and attribute data to you. Because of that reason, the data you provide cannot be withdrawn after web forms are completed. Within the study, the following data will be collected:

- User-generated information: the inputted password and PIN code
- Technical environment information: device model (e.g. iPhone X), operating system (e.g. iOS 12.4), Internet browser version (e.g. Safari)
- Statistical information: time-stamps on user-system interactions such as taps, scrolls, button pushes, checkbox ticks

How will the collected data be handled and used?

Up to the end of the data collection phase 15.09.2024, the anonymous data will be stored securely in the cloud-based database, to which only the research author has access. In the data analysis phase, the anonymous data accrued in the previous step will be moved onto a researcher's local drive outside any network and inaccessible to a third party. Then, the results of the data analysis will be published as a part of the Master's dissertation.

The data will be stored and analysed in plain text. After the University checks are completed and the research project is marked, the data will be destroyed with no recovery option.

Informed consent.

Your consent to participate in the research must be fully informed and voluntary. You have the right and opportunity to ask any questions and clarify any concerns related to your participation in the given study before providing your consent. There is a contact section at the end of this document for you to address any related inquiries.

Can you withdraw your consent on participation and collected data?

The study design aims at collecting the data anonymously. This fact excludes the possibility of identifying participants by any means. Therefore, withdrawal after completion of the assigned study tasks is unavailable due to the impossibility of mapping the data.

How can you find out about the results of the study?

The study outcomes will be available starting from 1 October 2024. A brief presentation of the key findings will be published on the website [<https://research-project-main-7b7d74af6ebc.herokuapp.com/>] where the participants took part in the research and will be publicly available up to 31 December 2024.

Ethical approvals.

This study is performed in accordance with Edge Hill University Research Ethics Policy available at <https://www.edgehill.ac.uk/wp-content/uploads/documents/research-ethics-policy-1-1.pdf>.

The research proposal has been reviewed and then granted ethical approval by the Department of Computer Science of Edge Hill University.

Contact Details.

Researcher Name: Veronika Chupova
Researcher Email Address: 25868705@edgehill.ac.uk

Appendix B. Ethical Consideration List (1/3)

Computing Department Ethical Check List

Ethical considerations

Your research project is unlikely to result in any major ethical issues, but please read what follows in order to acquaint yourself with some of the current thinking about the ethics of doing research. Not all of what follows may be applicable to your research.

This document **MUST** be read in conjunction with the Universities [Research Guidance](#)

Ethical considerations are paramount. They need to be fully discussed with your tutor before setting out on your project. Any concerns which are relevant to your investigation should be noted and will constitute an important aspect of your final discussion. For example, particular issues concerning the involvement of children or vulnerable adults need to be thought through carefully; likewise, issues around discussion of sensitive topics; issues around intrusion, if observation strategies are employed, and so on. All these matters need to be addressed before you set about collecting data. Evidence of in-depth reflection on ethical issues should be clear in your final report.

There are certain generally accepted guidelines governing research practice of which you must be aware:

1. Consent

As much information as possible should be provided to participants so that they can give - or withhold - their agreement to participate. Establishing consent is not always a straightforward business and requires careful and perceptive handling.

2. Deception

Intentional deception of participants about the purpose and general nature of the investigation should normally be avoided. If your project involves withholding any information from your participants, you must discuss this in detail with your tutor before going ahead.

3. Debriefing

At the end of the study you should give participants any further information needed to complete their understanding of the nature of the research, what you hope to do with it, and how it might affect them personally at any later date.

4. Rights to withdraw from the investigation

Participants have the right to drop out of the study at any time and this must be made clear to them from the outset. Remember that participants also have the right to withdraw consent retrospectively and require that their data be destroyed. They should be informed about this and enabled to take appropriate action.

5. Confidentiality and/or anonymity.

These issues should be fully discussed with prospective participants. You need to be very clear about how 'confidentiality' and 'anonymity' are different, and what guarantees you will - and conversely will not - be able to give to participants in terms of respecting either, or both, of these.

6. Protection of participants from physical and mental harm during the investigation.

This is essential. It may appear to be a rather extreme consideration in relation to the small-scale research project, but you must think carefully about any levels of stress or distress which participation might cause for your participants either during or after the research.

Appendix B. Ethical Consideration List (2/3)

Computing Department Ethical Check List

Sensitivity to Multicultural Issues

When constructing interview schedules, questionnaires or other data-gathering instruments, it is important to be sensitive to different cultural perspectives. Depending upon the nature and location of your research, the following issues should be considered:

1. Your own prejudices and biases.
2. Though we may strive to be objective, we all have our own, often unacknowledged, prejudices. They may relate to a person's age, ethnicity, gender, sexual orientation, religion, disability, or marital, parental or socio-economic status.
3. Sensitivity to the language used in the research process to describe the different groups involved: e.g. Are explicitly derogatory terms used in describing children or adults from 'other' groups? Are subtly derogatory terms used, such as 'these people' when describing participants in a project?
4. Is the language used in data-collection instruments accessible and understandable to all participants?
5. Are members/representatives of all groups of participants involved in planning, implementing, and reviewing results from the research?
6. Have multicultural issues been addressed openly at all stages of the research?
7. Does the group of participants represent the cultural diversity of the institution/area? What implications may this have for the research findings?
8. Could the results of the research be viewed differently by different cultural groups? What has been done to ensure that their perspectives have been included?


Stage 1 Self-Assessment

Part A	
If your research involves human participants, are any of the following concerns relevant?	
No	1. The involvement of vulnerable participants or groups, such as children (under the age of 16), people with a learning disability or cognitive impairment, or persons in a dependent relationship?
No	2. The sensitivity of the research topic, e.g. the participants' sexual, political, or legal behaviour, or their experience of violence, abuse or exploitation?
No	3. The gender, ethnicity, language, or cultural status of participants?
No	4. The use of deception, trickery, or other procedures that may contravene participants' full or informed consent, without timely and appropriate debriefing, or activities that cause stress, humiliation, or anxiety, or the infliction of more than minimal pain?
No	5. Access to records of personal or other confidential information, including genetic or other biological information, concerning identifiable individuals without their knowledge or consent?
No	6. The use of intrusive interventions, such as the administration of drugs or other treatments, excessive physical exertion, or techniques such as hypnotherapy without the participants' knowledge or consent?
No	7. Research related to the NHS is strongly advised to seek advice from their supervisor before commencing the project

If you have answered 'Yes' to any of the questions then the project is considered to be of **high ethical risk** and may need to be approved by the Departments Ethics Committee.
Please discuss your project with your supervisor.

Appendix B. Ethical Consideration List (3/3)

Computing Department Ethical Check List

Part A	
If your research involves human participants, are any of the following concerns relevant?	
Otherwise, your project may be considered low ethical risk . Please sign below and submit your self-assessment document to your supervisor and upload it to BB.	
Approval for Low Risk Research Projects	
I can confirm that : <i>(confirm you have read these)</i>	
- I have read the Edge Hill University Framework for Research Ethics https://www.edgehill.ac.uk/document/research-ethics-policy/	YES
- I have read the Computing Departments Ethics Policy Document (see BB)	YES
- I agree to abide by their principles	YES
Your Signature ⁱ	
Your name	Veronika Chupova
Date:	21 June 2024
Supervisor's Signature ⁱⁱ	VINH TA
Supervisor's name	Vinh-Thong Ta
Date	21/06/2024

