# 3rd Factor Authentication

Venkata Hari Paniindra Inala, A Sai, Ch Abhiram and
Rashmi Prasad

December 18, 2024

# $3^{rd}$ Factor Authentication

**Bachelor of Technology**

**In**

**Computer Science and Engineering (Internet of Things)**

**Design & Developed by**

| | |
|---|---|
| I.Venkata Hari Paniindra | 2111CS050106 |
| A.Sai | 2111CS050095 |
| C. Abhiram | 2111CS050098 |

Under the Esteemed guidance

**Mrs. K. Rashmi**

Assistant Professor

**Department of Computer Science and Engineering**

**Internet of Things**

**School of Engineering**

**MALLA REDDY UNIVERSITY**

**Maisammaguda, Dulapally, Hyderabad,500100**

**2021-2025**

MALLA REDDY UNIVERSITY

(Telangana State Private Universities Act No.13 of 2020 and G.O.Ms.No.14, Higher Education (UE) Department)

# CERTIFICATE

This is to certify that this is the bonafide record of the application development entitled “**3rd factor authentication”**, submitted by **I. Venkata Hari Paniindra (2111CS050106), Ch. Abhiram(2111CS050098), S.Sai (2111CS050095).** B. Tech IV year I semester, Department of CSE (IoT) during the year 2024-25. The results embodied in this report have not been submitted to any other university or institute for the award of any degree or diploma.

| | |
|---|---|
| **Internal Guide** | **Head of Department** |
| **Mrs. K. Rashmi** | **Dr. G. Anand Kumar** |
| **Assistant Professor** | **CSE(IOT)** |

**External Examiner**

# DECLARATION

We hereby declare that the project report entitled "**3$^{rd}$ Factor Authentication**"has been carried out by us and this work has been submitted to the **Department of Computer Science and Engineering (Internet of Things), Malla Reddy University**, Hyderabad in partial fulfilment of the requirements for the award of degree of Bachelor of Technology. We further declare that this project work has not been submitted in full or part for the award of any other degree in any other educational institutions.

Place:

Date:

| | |
|---|---|
| I.Venkata Hari Paniindra | 2111CS050106 |
| A.Sai | 2111CS050095 |
| C.Abhiram | 2111CS050098 |

# ACKNOWLEDGEMENT

We would like to express our gratitude to all those who extended their support and suggestions to come up with this software. Special Thanks to our mentor **Mrs.K.Rashmi** whose help and stimulating suggestions and encouragement helped us all time in the due course project development.

We sincerely thank our Head of the Department **Dr. G. Anand Kumar** for his constant support and motivation all the time. A special acknowledgement goes to a friend who enthused us from the backstage. Last but not the least our sincere appreciation goes to our family who has been tolerant, understanding our moods and extending timely support.

# ABSTRACT

Third-factor authentication (3FA) enhances the security of digital systems by adding an additional layer of verification beyond traditional methods such as passwords and biometric identifiers. This authentication model typically incorporates a third factor, such as a physical token, smart card, or a one-time password (OTP) delivered via a separate communication channel, to further validate the identity of a user. By leveraging the increased complexity provided by a third factor, 3FA can significantly reduce the risk of unauthorized access, phishing attacks, and identity theft, making it an essential component in high-security application

# INDEX

# CHAPTER - 1

# INTRODUCTION

## 1.1 Problem Definition & Description

As the digital landscape evolves, attackers increasingly exploit weak links in authentication mechanisms to gain unauthorized access to sensitive systems. While two-factor authentication has become a standard, it often relies on physical devices or transient codes, which can be lost, intercepted, or compromised. Biometric methods, though secure, raise concerns about data privacy, irreversible breaches, and deployment challenges. In this context, word locks offer a novel approach to third-factor authentication.

The increasing sophistication of cyber attacks has exposed significant vulnerabilities in traditional authentication systems, which primarily rely on passwords (first factor) and tokens or OTPs (second factor). While biometrics serve as a common third-factor solution, they present challenges such as high costs, privacy concerns, and limited accessibility in certain environments. This underscores the need for an alternative third-factor authentication method that is secure, cost-effective, and user-friendly. A word lock, which requires users to recall a predefined sequence of words, offers a promising solution. By leveraging cognitive memorization, word locks provide an additional layer of security without the need for external devices or sensitive biometric data, making them a practical and innovative approach to enhancing authentication systems..

## Problem Statement

With the increasing reliance on digital platforms, ensuring the security of user authentication systems has become paramount. Traditional authentication methods, including passwords (first factor) and OTPs or hardware tokens (second factor), are no longer sufficient to safeguard against sophisticated cyberattacks such as phishing, credential theft, and social engineering. While biometric authentication is a common third-factor solution, it raises concerns regarding privacy, cost, and implementation complexity. Hence, there is a pressing need for an alternative third-factor authentication method that is secure, cost-effective, user-friendly, and compatible with existing systems.

## 1.2 Objective of the Project

The objectives of cold storage monitoring system include:

1. **Enhance Authentication Security**

To develop a secure third-factor authentication mechanism using word locks to complement existing first and second-factor methods.

2. **Improve Usability**

   To design a system that is user-friendly, enabling easy recall and operation without the need for external devices or complex tools.

3. **Ensure Compatibility**

   To integrate the word lock authentication mechanism seamlessly with existing authentication frameworks and digital platforms.

4. **Address Privacy Concerns**

   To offer an alternative to biometrics that avoids the storage of sensitive personal data, reducing privacy risks.

5. **Promote Cost-Effectiveness**

   To create an affordable and practical solution suitable for resource-constrained environments, eliminating the need for expensive hardware or software.

6. **Increase Accessibility**

   To provide a third-factor authentication method that can be widely adopted across diverse demographics, including those with limited technological resources.

7. **Mitigate Common Security Risks**

   To reduce vulnerabilities associated with password theft, phishing, and device loss by introducing a robust layer of cognitive-based authentication.

8. **Encourage Scalability and Future Growth**

   To lay the groundwork for future enhancements in authentication systems by leveraging the versatility of the word lock concept.

## 1.2 Scope of the project

The project focuses on designing and implementing a word lock mechanism as a third-factor authentication system to enhance the security of digital platforms. This innovative approach aims to address the limitations of existing authentication methods by providing a secure, user-friendly, and cost-effective solution.

# CHAPTER - 2
# SYSTEM ANALYSIS

## 2.1 Existing System

### 2.1.1 Background & Literature Survey

Third-factor authentication (3FA) is designed to provide an additional layer of security by incorporating "something you are" (e.g., biometrics) or alternative methods. While biometrics are effective, they pose challenges related to privacy, cost, and accessibility. This has led to the exploration of innovative, cognitive-based solutions like word locks. By requiring users to recall and input a predefined sequence of words, word locks address key limitations of existing systems, offering a user-friendly, cost-effective, and secure third-factor authentication method

### 2.1.2 Design Principles of Cold Storage Systems

The design of a third-factor authentication (3FA) system must prioritize security, ensuring it provides an additional layer of protection against common threats like phishing, credential theft, and brute-force attacks. Simplicity and usability are crucial, allowing users to easily interact with the system without extensive technical knowledge. The system should also protect user privacy by minimizing the storage and transmission of sensitive data, especially when compared to biometric methods. It must be compatible with existing authentication frameworks and cost-effective to ensure broad adoption

### 2.1.2 Limitations of Existing System

The limitation of password-based systems are vulnerable to phishing, credential theft, and brute-force attacks, while two-factor authentication (2FA) methods relying on external devices or networks are prone to being lost or compromised. Biometric systems, though secure, raise privacy concerns and are expensive to implement, with the risk of sensitive data breaches. Additionally, complex authentication methods can create usability challenges, and systems like SMS-based 2FA are susceptible to interception and SIM swapping.

## 2.2 Proposed System

The proposed system introduces a word lock mechanism as a third-factor authentication (3FA) solution, designed to enhance security while addressing the limitations of existing authentication methods. The word lock requires users to recall and input a predefined sequence of words, adding an extra layer of cognitive-based security. This method leverages the human ability to remember sequences of words, making it more secure than traditional passwords without relying on external devices or sensitive biometric data.

### 2.2.1 Advantages of proposed System

1. **Enhanced Security**
   The word lock system adds an extra layer of security by requiring a third factor of authentication, reducing the risk of unauthorized access from common threats like phishing, password theft, and brute-force attacks.
2. **User-Friendly**
   It leverages the human ability to recall words, making it easier for users to remember and input their credentials without the complexity of traditional passwords or biometric systems.
3. **Cost-Effective**
   Unlike biometric systems or hardware tokens, the word lock mechanism does not require expensive infrastructure, making it an affordable solution for both users and organizations.
4. **No External Device Dependency**
   The system does not rely on external devices (such as hardware tokens, mobile phones, or biometric scanners), eliminating the risks associated with lost or compromised devices.
5. **Privacy Protection**
   Since the system does not collect or store sensitive personal data, such as biometric information, it minimizes privacy concerns and the risk of data breaches.

## 2.3 Software & Hardware Requirements

### 2.3.1 Software Requirements

Apache , Node.js , HTML , Java script , CSS , Mongo DB, Express.js

### 2.3.2 Hardware Requirements

Server , Computing Device , Storage Device , Network Equipment , User Device

## 2.4 Feasibility Study

### 2.4.1 Technical Feasibility

The technical feasibility of implementing the word lock-based third-factor authentication (3FA) system is high due to the use of existing, widely adopted technologies such as web development frameworks (HTML, CSS, JavaScript), backend programming languages (Python, Java, Node.js), and secure database management practices. The system can easily integrate with current authentication frameworks like Multi-

Factor Authentication (MFA) and Single Sign-On (SSO), leveraging established APIs and libraries for seamless integration.

### 2.4.2 Robustness & Reliability

The word lock-based third-factor authentication (3FA) system is designed to be both robust and reliable, ensuring consistent performance and security across various use cases. Robustness is achieved by utilizing well-established encryption methods and security protocols, which safeguard against common vulnerabilities such as phishing, credential theft, and brute-force attacks. The system's reliance on a user-defined word sequence adds an additional layer of security without compromising usability, ensuring that the authentication method remains resistant to unauthorized access attempts. Furthermore, the word lock system does not depend on external devices or networks, minimizing the risk of failure due to device loss, network outages, or hardware malfunctions . Reliability is ensured through the use of secure and scalable infrastructure. The system can be deployed on cloud platforms with automatic failover and load balancing, making it resilient to traffic spikes and hardware failures. Redundancy and secure backup protocols further enhance its reliability, ensuring minimal downtime and uninterrupted access. The system also supports continuous monitoring, allowing quick detection and resolution of issues, which contributes to overall uptime and system health. Additionally, because it integrates seamlessly with existing authentication frameworks, it ensures consistency and reliability across different environments, whether for individual users or enterprise-level deployments.

### 2.4.3 Economic Feasibility

The systems offer various economic benefits by optimizing energy usage, reducing product losses, enhancing operational efficiency, and minimizing maintenance costs. This section examines the economic feasibility of implementing such systems:

1 **Low Initial Costs**
The system uses existing technologies (web development frameworks, backend languages, and secure databases), reducing development and implementation costs.

2 **No Need for Expensive Hardware**
Unlike biometric or hardware token-based systems, the word lock system does not require specialized devices, making it more affordable to deploy and maintain.

3 **Utilizes Standard Devices**
The system relies on common computing devices (PCs, smartphones, tablets), which users already have, avoiding additional hardware costs.

4 **Scalable Infrastructure**

The system can be hosted on cloud platforms like AWS, Google Cloud, or Microsoft Azure, which offer pay-as-you-go pricing, reducing capital expenditure and aligning costs with actual usage.

5 **Reduced Maintenance Costs**

Minimal ongoing maintenance is required compared to hardware-based solutions, reducing long-term operational costs.

6 **Low Security Implementation Costs**

Security features such as encryption and hashing are standard, low-cost practices in modern authentication systems.

7 **Cost Savings from Improved Security**

By enhancing security and reducing vulnerabilities, the system helps prevent costly data breaches, protecting organizations from financial losses and reputational damage.

# CHAPTER – 3
# ARCHITECTURAL DESIGN

## 3.1 Module Design

The module design for the word lock-based third-factor authentication (3FA) system is organized into distinct components, each responsible for specific functions within the authentication process. The modular approach ensures flexibility, maintainability, and scalability. Below are the key modules:

1. **User Interface (UI) Module**
   - Function: Provides a user-friendly interface for entering credentials, including the word lock sequence, username, and possibly other factors such as passwords or OTPs.
   - Components:
     - Login Form: A simple form for users to enter their username and word lock sequence.
     - Feedback System: Displays messages to users based on successful or failed authentication attempts, guiding them through the process.
   - Technologies Used: HTML, CSS, JavaScript (React, Angular, or Vue.js for dynamic elements).
2. **Authentication Module**
   - Function: Responsible for processing user input, validating the word lock, and verifying it against stored credentials.
   - Components:
     - Input Capture: Collects the word lock and other factors (username, password) from the UI.
     - Word Lock Validation: Hashes the entered word lock sequence and compares it with the stored hash from the database.
     - Authentication Logic: Integrates the word lock with any other authentication factors (password, OTP) as part of the 3FA process.
   - Technologies Used: Server-side languages (Python, Java, Node.js), bcrypt or PBKDF2 for hashing, secure password management libraries.
3. **Database Module**
   - Function: Stores and retrieves user credentials securely, including the word lock sequence, password, and other related data.

- Components:
    - User Data Storage: Stores user credentials, including the hashed word lock sequences and passwords.
    - Encryption and Hashing: Ensures that sensitive data is securely stored using encryption and hashing techniques (e.g., bcrypt, PBKDF2).
    - Audit Logs: Records authentication attempts, successful logins, and potential security breaches.
- Technologies Used: Relational databases (MySQL, PostgreSQL) or NoSQL databases (MongoDB), encryption libraries.

4. **Security Module**

- Function: Protects user data and the authentication process through encryption, secure communication, and session management.
- Components:
    - Encryption: Implements SSL/TLS for encrypted communication between the client and server (HTTPS).
    - Session Management: Issues and manages secure tokens (e.g., JWT) for authenticated users to ensure secure access to protected resources.
    - Password Hashing: Uses secure algorithms (bcrypt, PBKDF2) to hash and salt passwords and word lock sequences.
- Technologies Used: SSL/TLS for HTTPS, JWT or session cookies for session management, bcrypt/PBKDF2 for hashing.

5. **Multi-Factor Authentication (MFA) Module**

- Function: Integrates additional layers of authentication (e.g., OTP or biometric) to strengthen the 3FA process.
- Components:
    - OTP Generation: Generates one-time passwords and sends them to users via SMS or email for verification.
    - Biometric Integration (optional): Can be added to integrate biometric data (e.g., fingerprint, facial recognition) as part of the authentication process.
- Technologies Used: SMS/Email APIs (Twilio, SendGrid), OTP generation libraries, biometric SDKs.

6. **Logging and Monitoring Module**

- Function: Monitors system activity, tracks authentication attempts, and generates logs for auditing and troubleshooting.
- Components:
  - Log Generation: Tracks all login attempts, failed authentication, and potential security threats.
  - Alert System: Notifies administrators of suspicious activity or security breaches (e.g., multiple failed login attempts).
- Technologies Used: Logging frameworks (e.g., Log4j, Winston), monitoring tools (e.g., Prometheus, ELK Stack).

## 3.2 Method & Algorithm design

1 **User Registration**

**Step 1**: User provides a unique username and selects a secure word lock sequence during the registration process.

**Step 2**: The word lock sequence is hashed using a secure hashing algorithm (e.g., bcrypt or PBKDF2).

**Step 3**: The hashed word lock sequence, along with other user details (like username and email), is securely stored in the database.

**Step 4**: The user also sets up other authentication factors (e.g., password, email address, or phone number for OTPs) as part of the multi-factor authentication (MFA) setup.

2 **User Authentication**

**Step 1**: The user enters their username and word lock sequence on the login page.

**Step 2**: The server receives the entered credentials and hashes the word lock sequence input.

**Step 3**: The server compares the hashed input against the stored hash for the word lock sequence in the database.

**Step 4**: If the word lock sequence matches, the system checks for other factors of authentication (e.g., password or OTP).

**Step 5**: If all factors are validated, the user is authenticated and granted access. A session token (e.g., JWT) is created to maintain the authenticated session.

3 **Multi-Factor Authentication (MFA)**

**Step 1**: After validating the word lock, the user may be prompted to enter a second factor of authentication, such as a password or a one-time password (OTP).

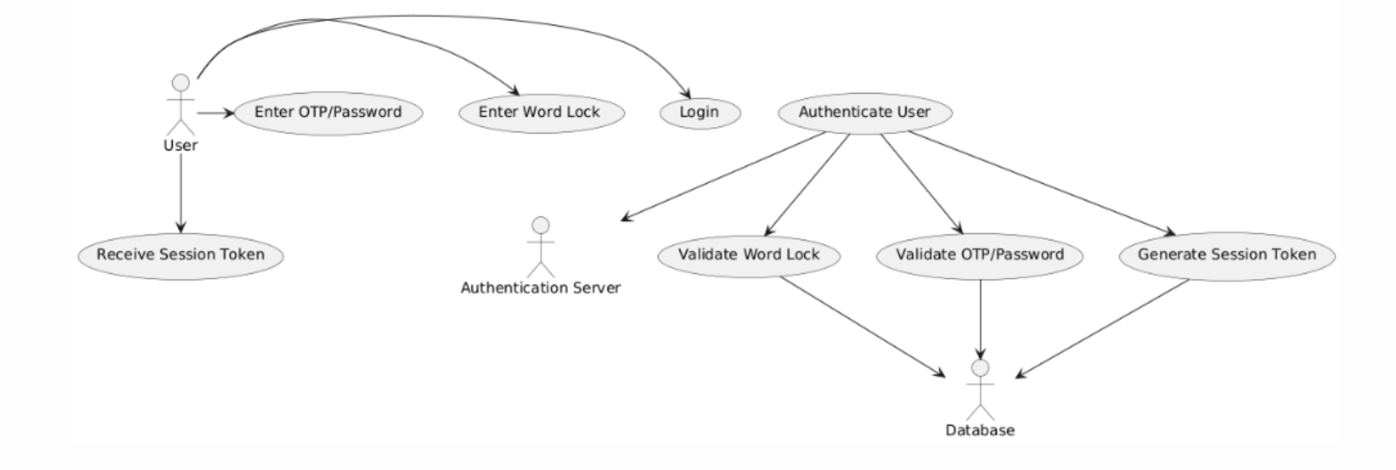
**Step 2**: For OTP, the system generates a unique code and sends it via SMS or email to the user.

**Step 3**: The user enters the OTP, and the server verifies it against the generated code.

**Step 4**: If the OTP matches and other factors are validated, the user is successfully authenticated.

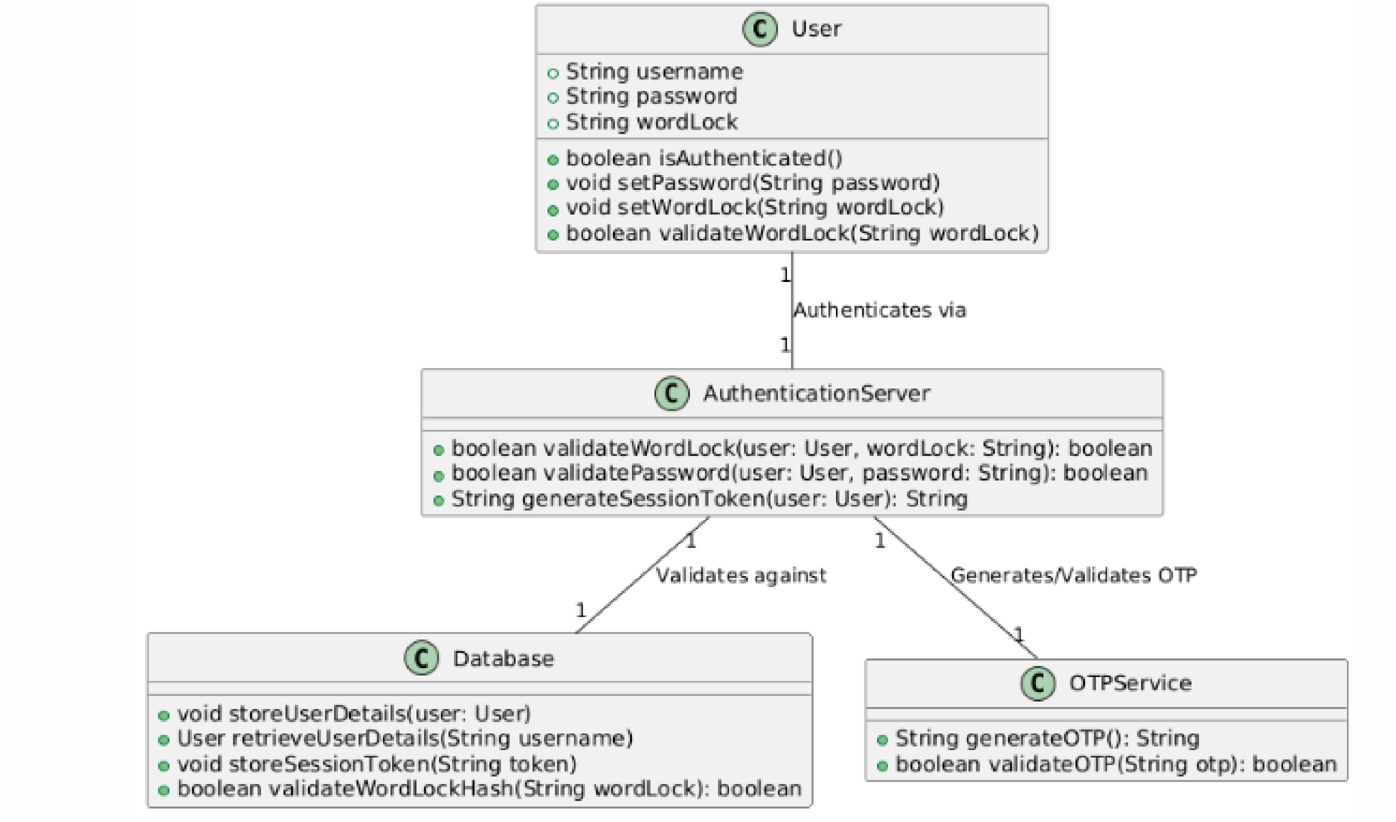
## 3.3 Project Architecture

### 3.3.1 Data Flow Diagram



- **User Device (Client Interface)**:The user device is where the interaction with the system takes place. It could be a smartphone, laptop, or any device used to access the application or service. The user enters their credentials (username, word lock sequence, and password/OTP) to initiate the authentication process.
- **Authentication Server**:The authentication server is responsible for verifying the identity of the user by validating their word lock, password, or OTP. It acts as the core logic that authenticates users and generates session tokens.
- **Database**: The database is where the user's data, including credentials (hashed word lock, password) and session tokens, are stored securely. It is a critical component for ensuring that the system can authenticate users properly by checking the stored information against the provided data.
- **Word Lock**: The word lock is a sequence of words or characters that serve as the "third factor" of authentication. The user must enter the correct sequence of words as part of the authentication process.

### 3.3.2 Class Diagram


User
String username
String password
String wordLock
boolean isAuthenticated()
void setPassword(String password)
void setWordLock(String wordLock)
boolean validateWordLock(String wordLock)
1
Authenticates via
1
AuthenticationServer
boolean validateWordLock(user: User, wordLock: String): boolean
boolean validatePassword(user: User, password: String): boolean
String generateSessionToken(user: User): String
1
Validates against
1
1
Generates/Validates OTP
1
Database
void storeUserDetails(user: User)
User retrieveUserDetails(String username)
void storeSessionToken(String token)
boolean validateWordLockHash(String wordLock): boolean
OTPService
String generateOTP(): String
boolean validateOTP(String otp): boolean

**User Class**:

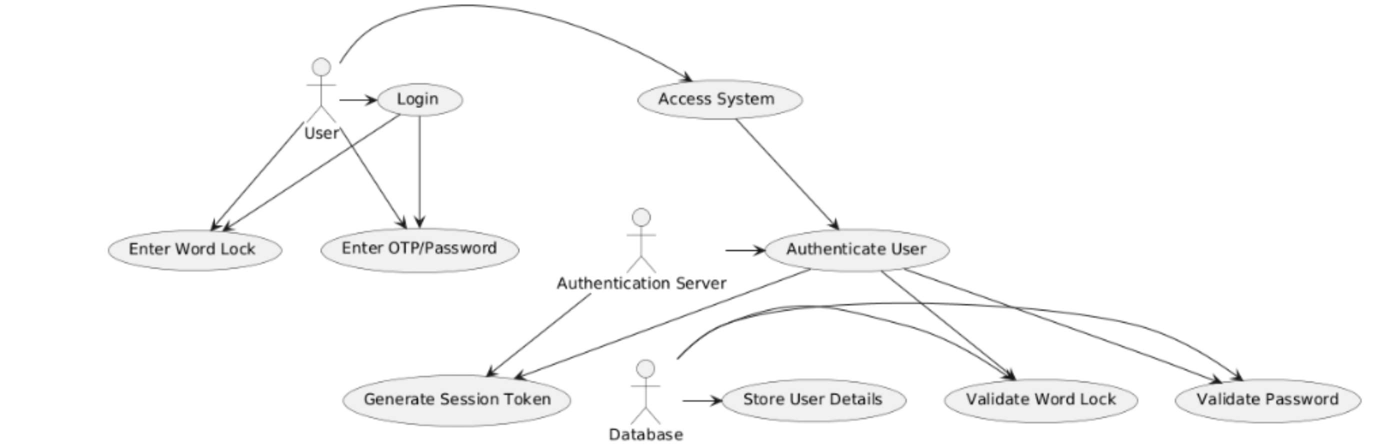
- **Attributes**:
  - username: Stores the username of the user.
  - password: Stores the password of the user.
  - wordLock: Stores the word lock sequence chosen by the user.
- **Methods**:
  - isAuthenticated(): Returns a boolean indicating whether the user is authenticated.
  - setPassword(String password): Sets the user's password.
  - setWordLock(String wordLock): Sets the user's word lock sequence.
  - validateWordLock(String wordLock): Validates the entered word lock sequence against the stored sequence.

### Authentication Server Class:

- **Methods:**

- validateWordLock(user: User, wordLock: String): boolean: Validates the word lock entered by the user.
- validatePassword(user: User, password: String): boolean: Validates the password entered by the user.
- generateSessionToken(user: User): String: Generates a session token after successful authentication.

### 3.3.3 Use Case Diagram



Use case Diagram

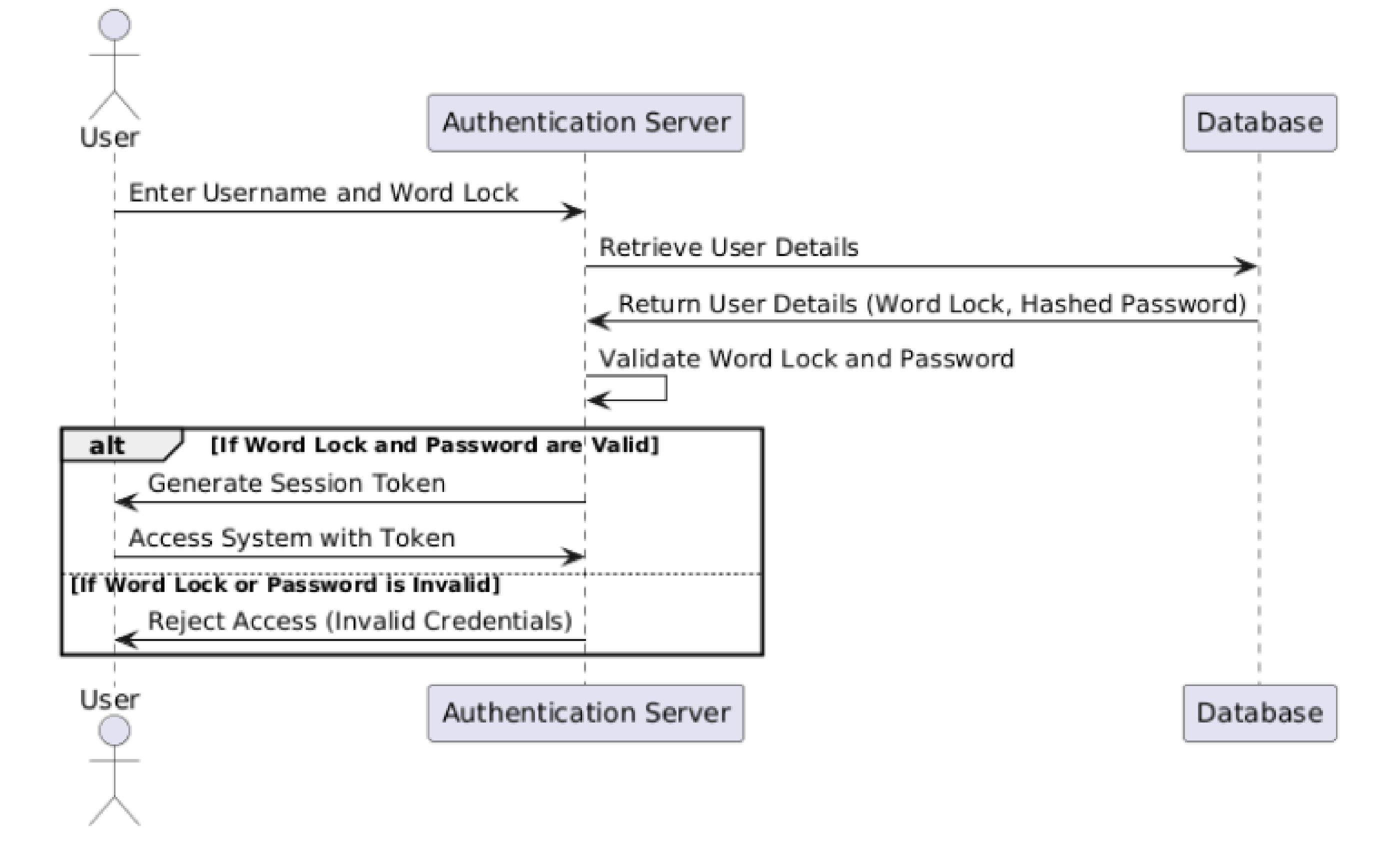
**Use Cases**:

- **Login**: The user initiates the login process.
- **Enter Word Lock**: The user enters the word lock sequence as part of the authentication process.
- **Enter OTP/Password**: The user enters the password or OTP for secondary authentication.
- **Access System**: After successful authentication, the user gains access to the system.
- **Authenticate User**: The authentication server validates the user's credentials.
- **Generate Session Token**: The authentication server generates a session token after successful authentication.
- **Store User Details**: The database stores the user's credentials (word lock, password).

- **Validate Word Lock**: The database validates the word lock sequence entered by the user.

- **Validate Password**: The database validates the password entered by the user.

### 3.3.5 Sequence Diagram


User
Authentication Server
Database
Enter Username and Word Lock
Retrieve User Details
Return User Details (Word Lock, Hashed Password)
Validate Word Lock and Password
alt
[If Word Lock and Password are Valid]
Generate Session Token
Access System with Token
[If Word Lock or Password is Invalid]
Reject Access (Invalid Credentials)
User
Authentication Server
Database

 **Actors**:

- **User**: The person trying to authenticate by providing their credentials (username, word lock, password, and OTP).

- **Authentication Server**: The component that validates the user's credentials and generates session tokens.

- **Database**: Stores the user's credentials and validates them during authentication.

- **OTP Service**: Provides OTP generation and validation functionality.

 **Sequence of Actions**:

- **User to Authentication Server**: The user begins the process by entering their username and word lock.

- **Authentication Server to Database**: The server retrieves the user's stored details (including word lock and hashed password) from the database.

- **Database to Authentication Server**: The database returns the user details.

- **Authentication Server**: The server then validates the entered word lock and password against the stored data.

- **Validation Outcomes**:

  - If both the word lock and password are valid, the server proceeds to generate an OTP.

  - The **OTP Service** generates an OTP, which is sent to the **Authentication Server**.

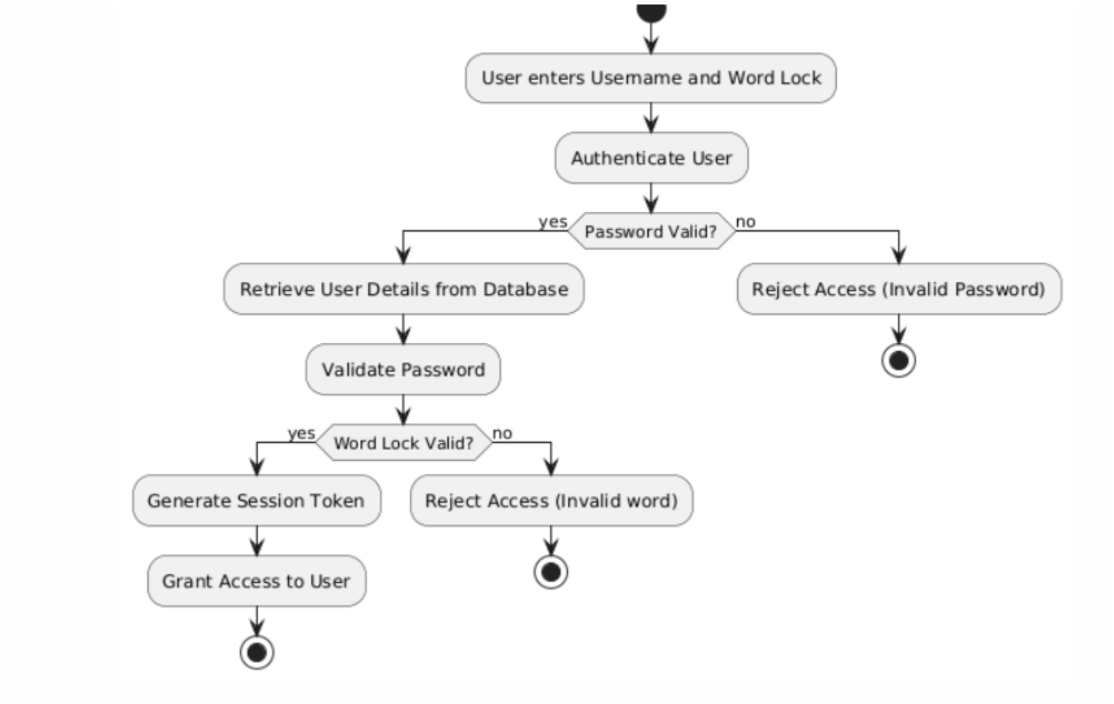
  - The **User** enters the OTP, which is then validated by the **OTP Service**.

  - If the OTP is valid, the **Authentication Server** generates a session token and grants the user access to the system.

  - If the OTP is invalid, the user is denied access.

- **Alternative Flow**: If either the word lock or password is invalid, the user is rejected immediately.

## 3.3.6 Activity Diagram


User enters Usemame and Word Lock
Authenticate User
Password Valid?
yes
no
Retrieve User Details from Database
Reject Access (Invalid Password)
Validate Password
Word Lock Valid?
yes
no
Generate Session Token
Reject Access (Invalid word)
Grant Access to User

## Flow of the Activity Diagram:

- The diagram follows a sequential process where the user is authenticated in steps: first with the word lock, then the password, followed by OTP validation.

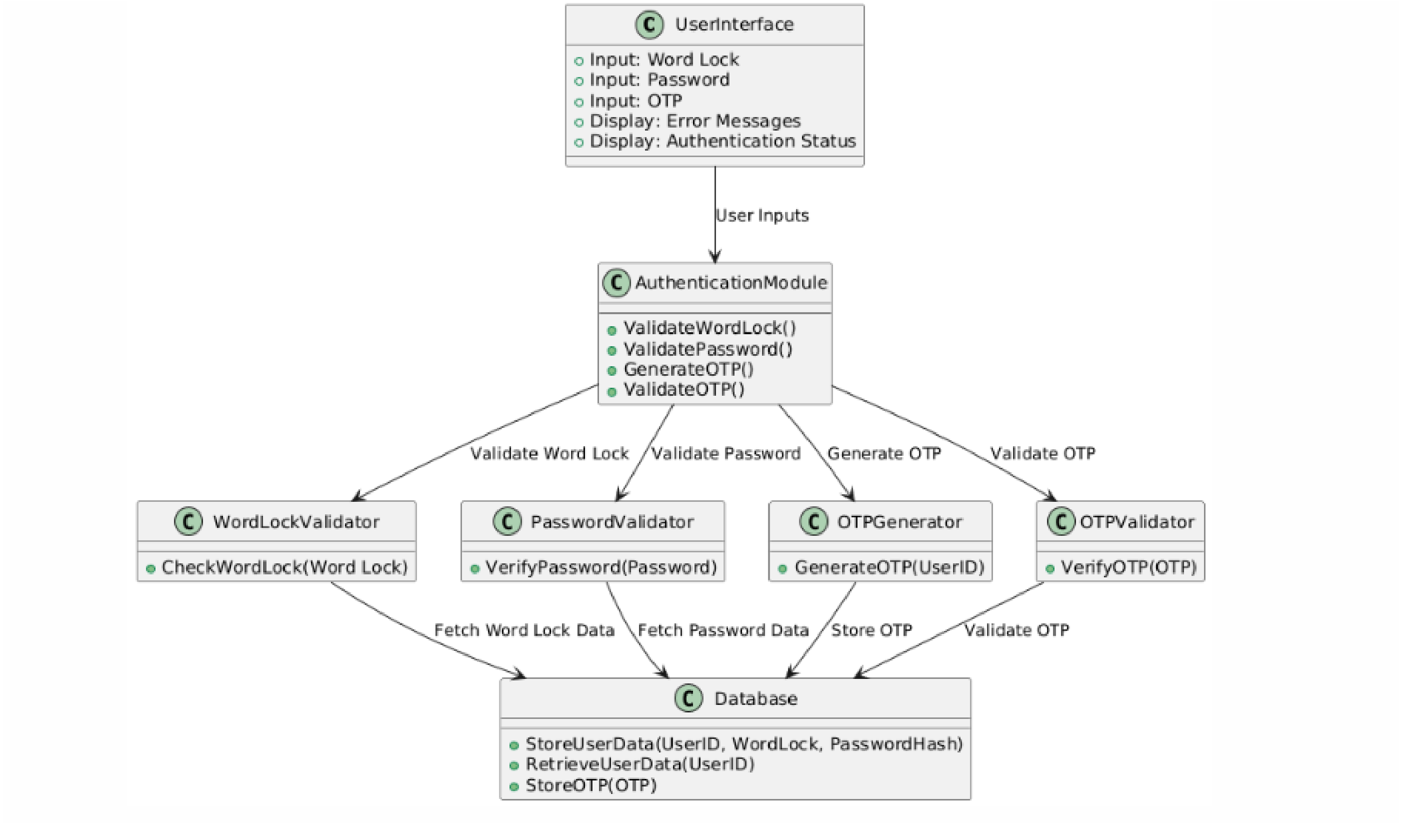
- If at any stage the credentials are invalid (word lock, password, or OTP), access is denied, and the process stops.

- If all validations pass, a session token is generated, and the user gains access.

# Chapter 4

# Implementation & Testing

## 4.1 Code Block



### Code flow

- The user initiates the process by entering a username and word lock.
- The Authentication Server validates the word lock and retrieves user details from the Database.
- If the word lock is valid, the user proceeds to enter their password.
- If both the word lock and password are valid, the system generates an OTP.
- The user enters the OTP, and the system validates it.
- Upon successful OTP validation, the system grants access and generates a session token.
- At any stage, if validation fails, the system rejects access and terminates the process.

## 4.2 Code execution

1. **Start:**
   - The user initiates the login process by providing their username and word lock.
2. **Step 1**: Word Lock Validation:
   - The Authentication Server receives the entered word lock and username.
   - It retrieves the corresponding user details (stored word lock and credentials) from the Database.
   - The entered word lock is compared with the stored word lock.
   - Decision:
     - If the word lock matches, proceed to the next step.
     - If not, reject the user and terminate the process.
3. **Step 2**: Password Validation:
   - The server prompts the user to enter their password.
   - The entered password is hashed and compared with the stored hashed password in the database.
   - Decision:
     - If the password is correct, proceed to the next step.
     - If not, reject the user and terminate the process.
4. **Step 3**: OTP Generation and Delivery:
   - If the password is valid, the Authentication Server generates a One-Time Password (OTP).
   - The OTP Service sends the OTP to the user via a pre-registered channel (e.g., SMS or email).
5. **Step 4:** OTP Validation:

- The user enters the received OTP on the authentication screen.
- The server verifies the OTP by checking it with the one generated and stored temporarily.
- Decision:
    - If the OTP matches, proceed to grant access.
    - If not, reject the user and terminate the process.

6. **Step 5:** Session Token Generation:
    - Upon successful OTP validation, the server generates a session token.
    - The session token is provided to the user, allowing secure access to the system.
7. **End:**
    - The user is granted access if all three authentication factors (word lock, password, and OTP) are valid.
    - If any of the factors fail, the process terminates with an access denied message.

## 4.3 Testing

### 4.3.1 Word Lock Validation

The first layer of the authentication process involves validating the user's word lock. Test cases in this category ensure that the system correctly identifies valid and invalid word locks. For example, when a user enters the correct word lock associated with their username, the system should successfully retrieve the user's data and proceed to password validation. Conversely, invalid word locks or special character inputs should trigger appropriate error messages, such as "Invalid word lock" or "Word lock cannot be empty." This ensures only legitimate users proceed beyond the first stage.

### 4.3.2 Password Validation

Once the word lock is verified, the system tests for password accuracy. Test cases here cover scenarios such as correct password inputs, incorrect passwords, and empty password fields. These tests ensure that passwords are securely hashed and compared against the database records. A valid password allows the user to proceed to OTP generation, while invalid or empty inputs prompt error messages or deny access. Tests also include cases where the password length exceeds the system's maximum limit, ensuring robustness against abnormal inputs.

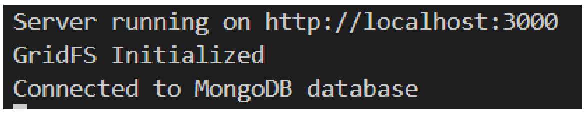
### 4.3.3. End-to-End Authentication

End-to-end tests validate the overall workflow by combining all three factors of authentication. These tests simulate real-world scenarios where users enter valid or invalid credentials at each stage. For instance, test cases ensure that a user with a valid word lock, password is granted access, while any failure in these components results in denied access. This ensures the system operates seamlessly and securely under normal and exceptional conditions.

# CHAPTER 5

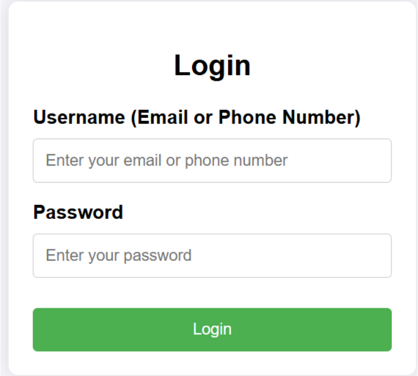
# RESULTS

## 5.1 Resulting Screens:



```
Server running on http://localhost:3000
GridFS Initialized
Connected to MongoDB database
```

### 5.1.1 Screen Shot 1 (Code Executed)



Login

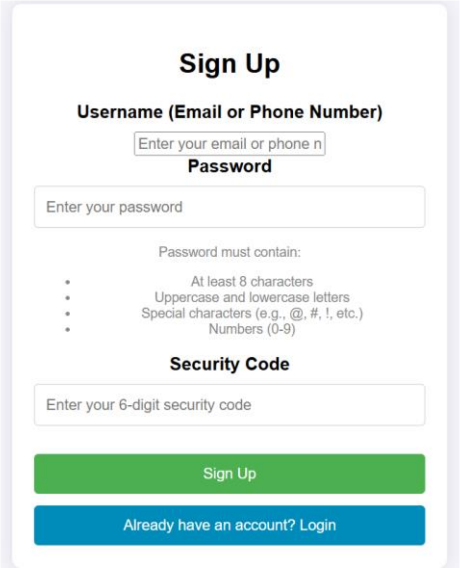
Username (Email or Phone Number)

Enter your email or phone number

Password

Enter your password

### 5.1.3 Screen Shot 3 Login Page



Sign Up

Username (Email or Phone Number)

Enter your email or phone n

Password

Enter your password

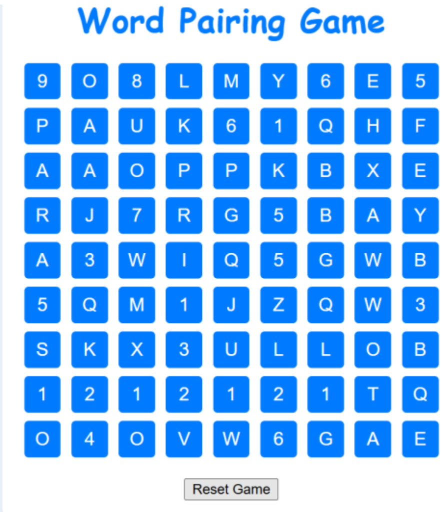
Password must contain:

- At least 8 characters
- Uppercase and lowercase letters
- Special characters (e.g., @, #, !, etc.)
- Numbers (0-9)

Security Code

Enter your 6-digit security code

### 5.1.2 Screen Shot 2 Sign Up page



Word Pairing Game

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 9 | O | 8 | L | M | Y | 6 | E | 5 |
| P | A | U | K | 6 | 1 | Q | H | F |
| A | A | O | P | P | K | B | X | E |
| R | J | 7 | R | G | 5 | B | A | Y |
| A | 3 | W | I | Q | 5 | G | W | B |
| 5 | Q | M | 1 | J | Z | Q | W | 3 |
| S | K | X | 3 | U | L | L | O | B |
| 1 | 2 | 1 | 2 | 1 | 2 | 1 | T | Q |
| O | 4 | O | V | W | 6 | G | A | E |

### 5.1.4 Screen Shot 4 Word Lock

## 5.2 Result Summary

The implementation and testing of the Word Lock-Based Third-Factor Authentication (3FA) system yielded promising results. The testing process evaluated various aspects of the system, including functionality, security, performance, and robustness, ensuring its effectiveness in real-world scenarios.

# CHAPTER 6

# CONCLUSION & FUTURE SCOPE

## 6.1 Conclusion

The Word Lock-Based Third-Factor Authentication (3FA) system successfully enhances traditional authentication methods by introducing an additional security layer rooted in user-defined word locks. This innovation addresses the limitations of existing systems, such as vulnerabilities to password breaches and insufficient user control over authentication factors. By integrating the word lock with standard password and OTP mechanisms, the system ensures a comprehensive and user-friendly approach to secure authentication.

## 6.2 Future Scope

The Word Lock-Based Third-Factor Authentication (3FA) system presents several opportunities for future enhancement and broader application. Its modular and scalable architecture allows it to evolve in response to emerging security challenges and technological advancements. Key areas for future development include:

1. **Integration with Biometric Authentication**:
   Combining the 3FA system with biometric factors such as fingerprints or facial recognition can create a more comprehensive multi-factor authentication (MFA) system, offering an unparalleled level of security.
2. **AI-Powered Threat Detection**:
   Incorporating machine learning algorithms to monitor and analyze login patterns can help identify and prevent potential threats, such as brute-force attacks or suspicious login behaviors.
3. **Cloud-Based Deployment**:
   Transitioning the system to a cloud-based platform would enhance scalability, allowing it to support a larger number of users and organizations while maintaining efficient performance.
4. **Cross-Platform Compatibility**:
   Expanding compatibility to include mobile, desktop, and IoT devices ensures broader usability and integration into diverse digital ecosystems.

## 6.3 Reference

1. National Institute of Standards and Technology (NIST). “Advanced Encryption Standard (AES)”. FIPS Publication 197. Available at: NIST .gov

2. Provos, N. & Mazieres, D. (1999). "A Future Adaptable Password Scheme." USENIX Annual Technical Conference. Available at: USENIX.org
3. Goldreich, O., Micali, S., & Wigderson, A. (1991). "Proofs that Yield Nothing but Their Validity or All Languages in NP Have Zero-Knowledge Proof Systems." Journal of the ACM. Available at: ACM Digital Library
4. Open Web Application Security Project (OWASP). “End-to-End Encryption Cheat Sheet.” Available at: OWASP.org
5. Cavoukian, A. (2009). “Privacy by Design: The 7 Foundational Principles.” Information and Privacy
6. European Union. "General Data Protection Regulation (GDPR)." Available at: GDPR .e