PBL PROJECT

On

One Time Password Generation using Javascript

Submitted to JNTU HYDERABAD

In Partial Fulfillment of the requirements for the Award of Degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted
By

N Harshitha	**218R1A0545**
P.Praharshini	**218R1A0551**
P Shravani	**218R1A0552**
V Sowjanya	**218R1A0562**

Under the Esteemed guidance of

Mr.K.VIJAYBABU

Assistant Professor, Department of CSE

Department of Computer Science & Engineering
CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(*Accredited by NBA*, Approved by AICTE, NEW DELHI, Affiliated to JNTU, Hyderabad)
Kandlakoya, Medchal Road, R.R. Dist. Hyderabad-501 401)

2023-2024

CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Accredited by NBA, Approved by AICTE, NEW DELHI, Affiliated to JNTU, Hyderabad)

Kandlakoya, Medchal Road, R.R Dist, Hyderabad-501 401)

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the project based learning entitled **“One Time Password Generation using Javascript”** is a bonafide work carried out by

N Harshitha	**218R1A0545**
P Praharshini	**218R1A0551**
P Shravani	**218R1A0552**
V Sowjanya	**218R1A0562**

in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING** from CMR Engineering College, affiliated to JNTU, Hyderabad, under our guidance and supervision. The results presented in this project based learning have been verified and are found to be satisfactory. The results embodied in this project based learning have not been submitted to any other university for the award of any other degree or diploma.

Internal Guide	Head of the Department
Mr.K.VIJAYBABU Assistant Professor Department of CSE CMREC, Hyderabad	**Dr. SHEO KUMAR** Professor & H.O.D Department of CSE CMREC, Hyderabad

DECLARATION

This is to certify that the work reported in the present project based learning entitled**“ One Time Password Generation using Javascript ”** is a record of bonafide work done by us in the Department of Computer Science and Engineering, CMR Engineering College, JNTU Hyderabad. The reports are based on the project based learning work done entirely by us and not copied from any other source. We submit our project based learning for further development by any interested students who share similar interests to improve the project based learning in the future.

The results embodied in this project based learning report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our knowledge and belief.

N Harshitha	**218R1A0545**
P Praharshini	**218R1A0551**
P Shravani	**218R1A0552**
V Sowjanya	**218R1A0562**

ACKNOWLEDGMENT

We are extremely grateful to **Dr. A. Srinivasula Reddy**, Principal and **Dr. Sheo Kumar**,HOD, **Department of CSE, CMR Engineering College** for their constant support**.** We are extremely thankful to **Mr. K.Vijaybabu ,** Assistant Professor, Internal Guide,Department of CSE, for his constant guidance, encouragement and moral support throughout the project based learning.We will be failing in duty if I do not acknowledge with grateful thanks to the authors of the references and other literatures referred in this Project based learning.

We express my thanks to all staff members and friends for all the help and co-ordinatio extended in bringing out this project based learning successfully in time.Finally, we are very much thankful to my parents who guided me for every step.

N Harshitha	**218R1A0545**
P Praharshini	**218R1A0551**
P Shravani	**218R1A0552**
V Sowjanya	**218R1A0562**

CONTENTS

TOPIC

ABSTRACT

A one-time password (OTP) is a temporary code that is valid for only one login session or transaction on a computer system or other digital device. It's often used as a second factor of authentication to enhance security,especially for sensitive accounts or transactions.Here's an abstract breakdown of how OTPs typically work:

OTPs are generated using algorithms that create unique codes based on a combination of factors such as time, a secret key, and a counter. Common algorithms include Time-based One-Time Password (TOTP) and HMAC-based One-Time Password (HOTP).OTPs can be delivered to the user through various channels including SMS, email, mobile apps, or hardware tokens. The delivery method chosen depends on the security requirements and user preferences. When the user attempts to log in or complete a transaction, they enter the OTP along with their username and password. The system then validates the OTP to ensure it matches the expected code generated for that particular session transaction.As the name suggests, OTPs are single-use only, meaning they can't be reused for subsequent login attempts or transactions. This adds an extra layer of security because even if the OTP is intercepted or stolen, it won't be useful for unauthorized access after its intended use.In TOTP-based OTP systems, the codes are time-sensitive and typically expire after a short period.

1.INTRODUCTION:

The introduction of an OTP (One-Time Password) generator marks a significant advancement in digital security, particularly in the realm of authentication and access control. An OTP generator is a tool or
system that dynamically produces temporary codes, which users must input alongside their regular credentials (such as a username and password) to verify their identity.

1.1 Importance of otp:

he importance of OTP (One-Time Password) in today's digital landscape cannot be overstated, as it serves as a crucial tool in enhancing security across various online platforms and transactions. Here are some key reasons why OTP is important:

1. **Strong Authentication**: OTP adds an extra layer of authentication beyond traditional username and password combinations. By requiring users to enter a temporary code along with their regular credentials, OTP ensures that only authorized individuals can access sensitive accounts or data.
2. **Mitigation of Password-based Attacks**: Password-based attacks, such as phishing, brute force, and dictionary attacks, are prevalent threats in cyberspace. OTP mitigates these risks by introducing dynamic, one-time codes that are valid for only a short duration, reducing the likelihood of successful unauthorized access even if passwords are compromised.
3. **Enhanced Security for Sensitive Transactions**: For online banking, e-commerce, and other sensitive transactions, OTP provides an additional layer of security to safeguard against fraudulent activities. It ensures that only legitimate users with access to their authorized devices can complete transactions, reducing the risk of financial losses and identity theft.
4. **Compliance Requirements**: Many regulatory standards and industry best practices mandate the use of OTP or multi-factor authentication (MFA) for securing sensitive data and systems. Compliance with these requirements not only helps organizations avoid penalties but also demonstrates a commitment to protecting user privacy and security.
5. **User Confidence and Trust**: Incorporating OTP into authentication processes instills confidence and trust among users, assuring them that their accounts and information are adequately protected. This sense of security can lead to higher user satisfaction and retention, as well as positive brand reputation.

6. **Adaptability to Various Platforms**: OTP can be implemented across a wide range of platforms and devices, including websites, mobile apps, email systems, and more. Its versatility makes it a flexible and scalable solution for securing diverse digital environments.
7. **Cost-Effective Security Measure**: Compared to other advanced security solutions, OTP is relatively simple and cost-effective to implement. Many OTP solutions are available as software-based applications or built into existing platforms, minimizing the need for expensive hardware or infrastructure upgrade.

1.2.Role in security enhancement:

OTP (One-Time Passwords) play a vital role in enhancing security across various digital platforms and transactions. Here are several key aspects highlighting their significance:

1. Two-Factor Authentication (2FA) or Multi-Factor Authentication (MFA): OTPs are often used as a second factor in 2FA or MFA systems. By requiring users to provide both something they know (e.g., a password) and something they have (e.g., a mobile device generating OTPs), OTP significantly reduces the risk of unauthorized access. This layered approach to authentication greatly enhances security.

2. Dynamic and Time-Sensitive Codes: OTPs are temporary codes that are valid for only a short duration, typically ranging from 30 seconds to a few minutes. Their dynamic nature ensures that even if intercepted, OTPs quickly become obsolete, minimizing the window of opportunity for attackers to misuse them. This time-sensitive feature enhances security by thwarting replay attacks and unauthorized access attempts.

3. Protection against Phishing and Credential Theft: OTPs provide a defense against phishing attacks and credential theft. Even if attackers manage to steal a user's password through phishing or other means, they would still need the OTP generated by the user's device to gain access. This added layer of security makes it significantly harder for attackers to compromise accounts.

4. Adaptive Security: OTPs can be adapted to suit different security needs and risk levels. Organizations can implement OTP solutions with varying levels of complexity, such as increasing the length of OTPs or requiring additional verification steps based on the sensitivity of the transaction or data being accessed. This adaptability allows for tailored security measures based on specific requirements.

5. Compliance with Regulatory Standards: Many regulatory standards and industry guidelines mandate the use of OTP or MFA for securing sensitive data and systems. Compliance with these standards not only helps organizations meet regulatory requirements but also strengthens overall security posture, reducing the risk of data breaches and regulatory penalties

2. SYSTEM ANALYSIS

2.1 Requirements gathering

Gathering requirements for implementing OTP (One-Time Password) systems involves a thorough understanding of the organization's security needs, operational constraints, and user expectations. Here's a structured approach to gathering requirements for OTPs:

1.Define Objectives and Scope:

- Identify the primary goals of implementing OTPs. Is it to enhance security, comply with regulatory requirements, or improve user experience?
- Define the scope of OTP implementation. Will it be used for user authentication, transaction verification, or both? Determine the specific systems or applications where OTPs will be required.

2.Identify Stakeholders:

- Identify key stakeholders involved in the OTP implementation process. This may include IT security personnel, system administrators, developers, compliance officers, and end users.
- Understand the perspectives and requirements of each stakeholder group. Consider their concerns, priorities, and expectations regarding OTP usage.

3.Assess Security Risks and Threats:

- Conduct a thorough risk assessment to identify potential security risks and threats that OTPs are intended to mitigate.
- Consider common threats such as phishing, credential theft, brute force attacks, and insider threats. Determine how OTPs can address these risks effectively.

4.Understand User Requirements:

- Gather user requirements through surveys, interviews, or user feedback sessions.
- Understand user preferences regarding OTP delivery methods (e.g., SMS, email, mobile app), frequency of OTP prompts, and usability considerations.
- Consider the usability needs of diverse user groups, including individuals with disabilities or limited access to technology.

2.2.Current System Assessment:

To provide a current system assessment for OTP (One-Time Password) systems, we need to evaluate several aspects:

1. Security Strength: OTP systems rely on generating unique passwords for each authentication attempt. The security strength depends on the algorithm used for generating these passwords (such as TOTP - Time-based OTP or HOTP - HMAC-based OTP), the length of the OTP codes, and the randomness of the codes generated. Generally, OTP systems are considered secure, especially when combined with other factors like username/password.

2. Implementation: The implementation of the OTP system can significantly impact its security. Vulnerabilities may arise from improper storage of secret keys, weak random number generation, or insecure transmission of OTPs. A thorough security review of the implementation is crucial.

3. Integration: OTP systems are often integrated with various platforms and applications for authentication purposes. The compatibility, ease of integration, and support for standard protocols (like OAuth, OpenID Connect) are essential considerations.

4. Usability: While security is paramount, usability is also crucial. OTP systems should be user-friendly, ensuring that users can easily generate or receive OTPs and input them during authentication without significant friction.

5. Scalability: The system should be able to handle a large number of OTP requests, especially in scenarios with high user volumes. Scalability considerations become critical, particularly for organizations with extensive user bases.

6. Backup and Recovery: Mechanisms for backup and recovery of OTP credentials should be in place to prevent lockout scenarios in case of device loss or failure.

3.SYSTEM DESIGN:

3.1Highlevel Architecture:

A high-level architecture for an OTP (One-Time Password) system typically involves several components working together to generate and validate one-time passwords for user authentication. Here's a basic outline of such an architecture:

1. User Interface: This is where the user interacts with the OTP system to request or input one-time passwords. The interface could be a web application, mobile app, or even SMS-based depending on the implementation.

2. Identity Provider (IDP): The IDP is responsible for managing user identities and authentication processes. It interfaces with the OTP system to trigger OTP generation and validation when users attempt to log in.

3. OTP Generator: This component generates one-time passwords based on a shared secret key and other parameters such as time (for TOTP - Time-based OTP) or a counter (for HOTP - HMAC-based OTP). The OTP generator typically uses cryptographic algorithms to ensure the uniqueness and randomness of the generated passwords.

4. Delivery Mechanism: Once generated, the OTP needs to be delivered to the user. This could be done through various channels such as SMS, email, push notifications, or generated within a mobile app.

5. OTP Validator: Upon receiving the OTP from the user, the OTP validator checks its validity by comparing it with the OTP generated based on the shared secret key and other parameters. If the OTP matches, the user is authenticated.

6. Shared Secret Storage: The secret key used for OTP generation and validation needs to be securely stored. This could involve using a secure key management system or encrypted databases.

7. Logging and Monitoring: To ensure security and track authentication attempts, logging and monitoring components are essential. They capture authentication events, failed attempts, and other relevant data for auditing and analysis purposes.

3.2.Componet design:

Designing the components for an OTP (One-Time Password) system involves breaking down the functionality into smaller, manageable parts, each responsible for specific tasks. Here's a detailed outline of the components for an OTP system:

1. User Interface (UI):

- **OTP Request Interface:** Allows users to initiate the OTP generation process.
- **OTP Input Interface:** Provides a way for users to input the OTP during the authentication

process.

2.Identity Provider (IDP):

- User Database: Stores user identities, including usernames, hashed passwords, and other relevant information.

- Authentication Service: Verifies user credentials and triggers the OTP generation process when required.

3. OTP Generator:

Algorithm Implementation: Implements the OTP generation algorithm (e.g., TOTP or HOTP).

Shared Secret Management: Handles the storage and retrieval of shared secrets associated with user accounts.

Time Synchronization (for TOTP): Ensures that the server's clock is synchronized with the client's clock for accurate OTP generation.

4. Delivery Mechanism:

SMS Gateway: Sends OTPs via SMS to users' registered phone numbers.

Email Service: Delivers OTPs via email to users' registered email addresses.

Push Notification Service: Sends OTPs as push notifications to users' mobile devices through dedicated mobile apps.

5. OTP Validator:

Validation Service: Verifies the OTP provided by the user against the OTP generated based on the shared secret and other parameters.

4.SOFTWARE & HARDWARE REQUIREMENTS:

4.1 Software requirements:

The software requirements for an OTP (One-Time Password) generator encompass various aspects to ensure the efficient and secure generation of OTPs. Here's a breakdown of the key software requirements:

1. Algorithm Implementation:

- Support for both TOTP (Time-based OTP) and HOTP (HMAC-based OTP) algorithms.
- Implementation of cryptographic functions required by the chosen OTP algorithm (e.g., SHA-1, SHA-256, HMAC).

2. Random Number Generation:

- Reliable and cryptographically secure random number generation to ensure the randomness of OTPs.
- Ability to generate random bytes for the generation of secret keys and OTPs.

3. Time Synchronization (for TOTP):

- Ability to synchronize with an accurate time source (e.g., NTP - Network Time Protocol) to generate time-based OTPs accurately.

4. Shared Secret Management:

- Secure storage and management of shared secrets associated with user accounts.
- Encryption of shared secrets at rest to prevent unauthorized access.

5. Configuration Options:

- Flexible configuration settings for parameters such as OTP length, time step interval (for TOTP), and counter (for HOTP).

6. Cross-Platform Compatibility:

- Support for multiple platforms, including desktop, web, and mobile environments.
- Availability of libraries or SDKs for popular programming languages (e.g., Python, Java, JavaScript) to facilitate integration into different applications.

7. Security Features:

- Protection against brute-force attacks through rate limiting and throttling mechanisms.
- Implementation of security best practices to mitigate vulnerabilities (e.g., secure coding practices, input validation, output sanitization).
- Support for encryption of communication channels to prevent interception of OTPs during transmission.

8. Integration Support:

- Ability to integrate with authentication systems, identity providers, and user directories for seamless user authentication workflows.
- Compatibility with standard protocols such as OAuth, OpenID Connect, and LDAP for interoperability with existing systems.

9. Logging and Auditing:

- Logging of OTP generation events, including timestamps, user identifiers, and generated OTPs.
- Support for auditing capabilities to track OTP usage and detect any suspicious activities.

10. Scalability and Performance:

- Ability to handle a high volume of OTP generation requests efficiently.
- Scalable architecture that can accommodate growing user bases and increased authentication loads.

4.2 Hardware requirements:

The hardware requirements for an OTP (One-Time Password) system depend on various factors including the expected load, security requirements, redundancy, and scalability needs. Here's a general outline of the hardware components typically involved in an OTP system:

1. Server Infrastructure:

- **Processing Power:** Sufficient CPU resources to handle the cryptographic operations required for OTP generation and validation.
- **Memory:** Adequate RAM to support concurrent user authentication sessions and caching of cryptographic keys.
- **Storage:** Sufficient disk space for storing configuration data, user accounts, shared secrets, and log files.

2. Network Infrastructure:

- **Network Bandwidth:** Sufficient network bandwidth to handle incoming authentication requests, especially during peak usage periods.
- **Redundant Network Connections:** Redundant network connections to ensure high availability and fault tolerance.
- **Firewalls and Intrusion Detection/Prevention Systems:** Hardware-based firewalls and IDS/IPS devices to protect the OTP system from unauthorized access and cyber threats.

3. Load Balancers:

- Hardware load balancers or Application Delivery Controllers (ADCs) to distribute incoming authentication requests across multiple OTP servers for load balancing and scalability.
- Load balancers also provide failover capabilities to redirect traffic in case of server failures.

4. Hardware Security Modules (HSMs):

- Dedicated HSMs for storing and managing cryptographic keys used in OTP generation and validation.
- HSMs provide tamper-resistant hardware-based security for protecting sensitive key material.

5. Redundancy and High Availability:

- Redundant server configurations with failover mechanisms to ensure uninterrupted service in case of hardware failures.
- Clustering or replication setups to replicate data and services across multiple servers or data centers for high availability.

6. Backup and Disaster Recovery:

- Backup solutions for regular backups of configuration data, user accounts, and cryptographic keys.
- Disaster recovery plans to recover the OTP system in case of catastrophic events such as hardware failures or natural disasters.

7. Monitoring and Management Tools:

- Hardware-based monitoring and management appliances for real-time monitoring of system health, performance metrics, and security events.
- Remote management capabilities for administering and troubleshooting the OTP system.

8. Power and Cooling:

- Uninterruptible Power Supplies (UPS) to provide backup power in case of power outages and ensure continuous operation.

- Adequate cooling systems to maintain optimal operating temperatures for hardware components.

9. Compliance and Security Measures:

- Hardware-based security mechanisms to comply with regulatory requirements and industry standards for data protection and confidentiality.

- Physical security measures to prevent unauthorized access to hardware components hosting sensitive data and services.

10. Scalability:

- Scalable hardware configurations that can be expanded or upgraded to accommodate growing user bases and increased authentication loads.
- Capacity planning to anticipate future growth and scale hardware resources accordingly.

5.IMPLEMENTATION:

5.1 Source code:

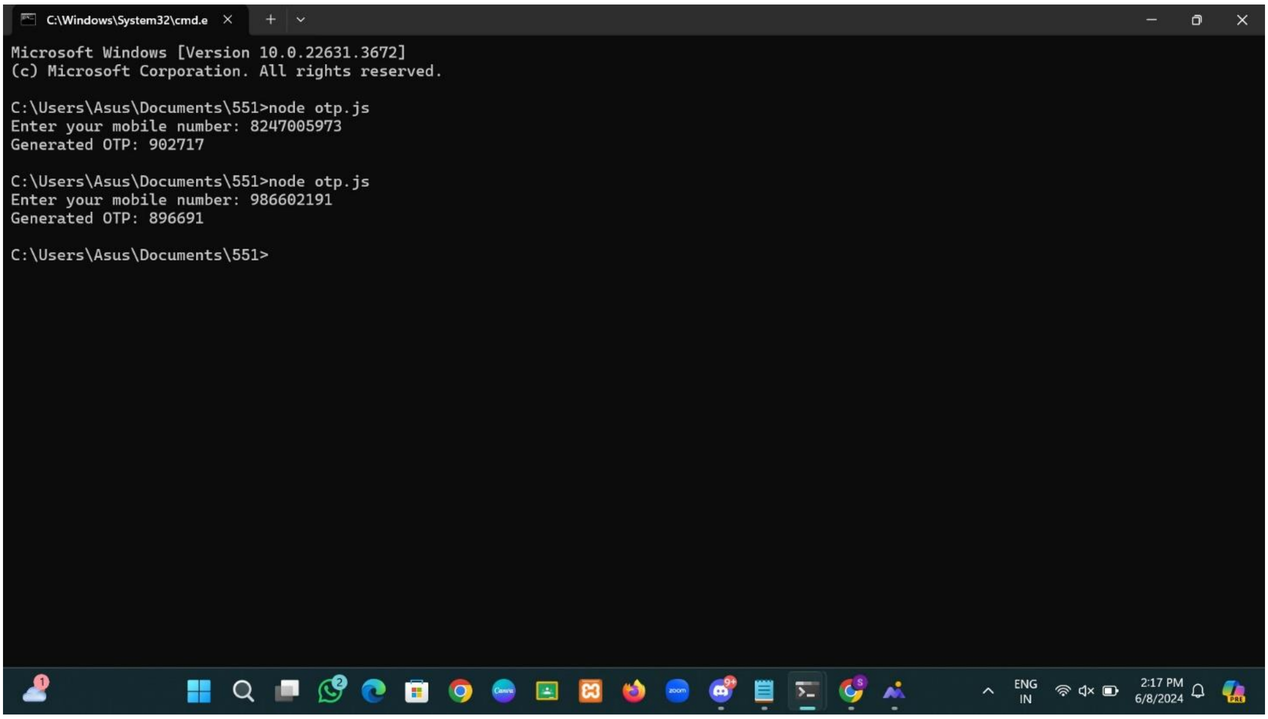
```
const readline = require('readline');
const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout
});

// Function to generate OTP
function generateOTP(mobileNumber) {
  // Generate a random 6-digit number
  const randomNumber = Math.floor(100000 + Math.random() * 900000);
  // Concatenate mobile number with random number
  const combinedStr = mobileNumber.toString() + randomNumber.toString();
  // Convert the string to an array of characters
  const charArray = combinedStr.split('');
  // Shuffle the array
  const shuffledArray = charArray.sort(() => 0.5 - Math.random());
  // Take the first 6 characters as OTP code
  const otpCode = shuffledArray.slice(0, 6).join('');
  return otpCode;
}

// Function to validate mobile number
function isValidMobileNumber(mobileNumber) {
  return /^\d{10}$/.test(mobileNumber); // Checks if the mobile number is exactly 10 digits
}

// Prompt the user to enter their mobile number
rl.question('Enter your mobile number: ', (mobileNumber) => {
  if (isValidMobileNumber(mobileNumber)) {
    const generatedOTP = generateOTP(mobileNumber);
    console.log(`OTP sent to ${mobileNumber}: ${generatedOTP}`);
// Now you can use the generated OTP for further validation or processing
        rl.close();
    } else {
        console.log('Invalid mobile number. Please enter a 10-digit number.');
        rl.close();
    }
});
```

6.OUTPUT:

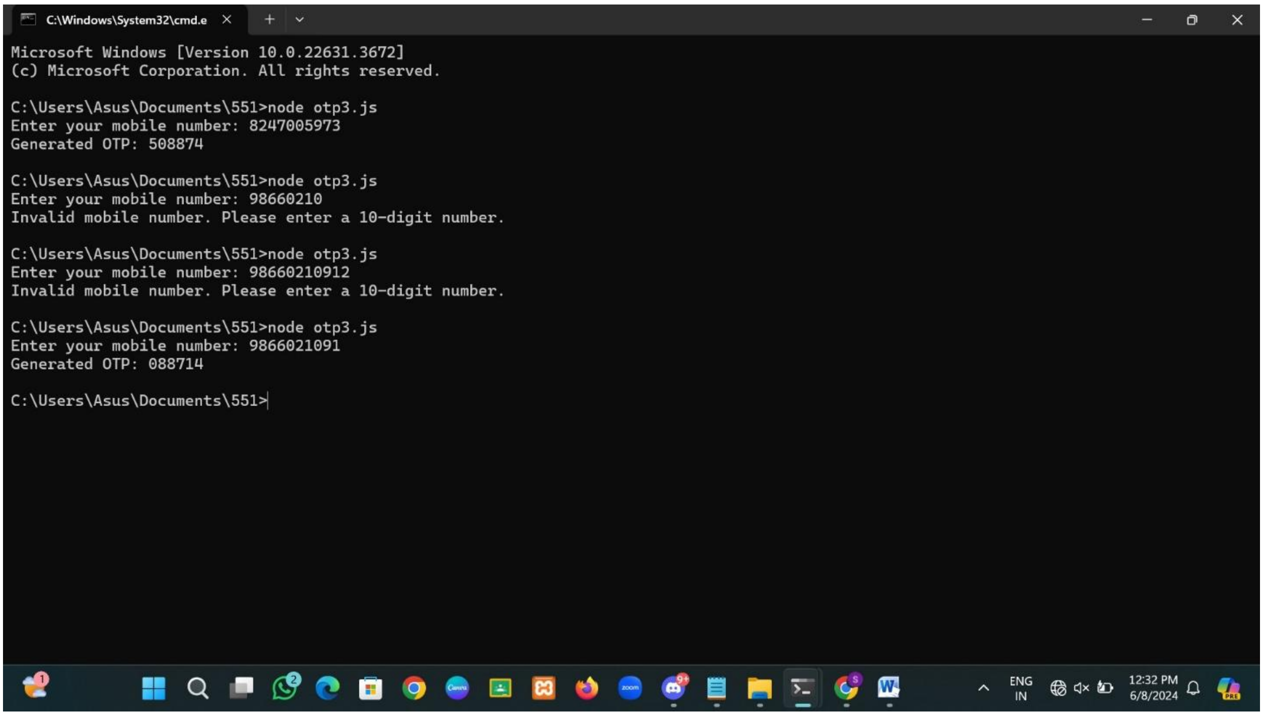


```
Microsoft Windows [Version 10.0.22631.3672]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Asus\Documents\551>node otp.js
Enter your mobile number: 8247005973
Generated OTP: 902717

C:\Users\Asus\Documents\551>node otp.js
Enter your mobile number: 986602191
Generated OTP: 896691

C:\Users\Asus\Documents\551>
```



```
Microsoft Windows [Version 10.0.22631.3672]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Asus\Documents\551>node otp3.js
Enter your mobile number: 8247005973
Generated OTP: 508874

C:\Users\Asus\Documents\551>node otp3.js
Enter your mobile number: 98660210
Invalid mobile number. Please enter a 10-digit number.

C:\Users\Asus\Documents\551>node otp3.js
Enter your mobile number: 98660210912
Invalid mobile number. Please enter a 10-digit number.

C:\Users\Asus\Documents\551>node otp3.js
Enter your mobile number: 9866021091
Generated OTP: 088714

C:\Users\Asus\Documents\551>
```

7.CONCLUSION:

In conclusion, OTP (One-Time Password) systems offer a secure and convenient method for authenticating users across various applications and platforms. Through the generation of unique passwords for each authentication attempt, OTPs enhance security by mitigating the risks associated with traditional static passwords.

Key aspects to consider when implementing an OTP system include:

1. Security Strength: OTP systems leverage cryptographic algorithms to generate unique passwords, ensuring a high level of security.

2. Implementation and Integration: Proper implementation and seamless integration with existing systems are crucial for the effectiveness of OTP systems.

3. Usability and Accessibility: While prioritizing security, OTP systems should also focus on user-friendliness to ensure smooth authentication experiences.

4. Scalability and Redundancy: As user bases and authentication loads grow, OTP systems should be scalable and redundant to maintain performance and availability.

5. Compliance and Standards: Adherence to regulatory requirements and industry standards is essential to protect user data and ensure compliance.

6. Monitoring and Maintenance: Continuous monitoring and regular maintenance are necessary to detect and address security threats and performance issues.

In summary, OTP systems provide a robust solution for enhancing authentication security, particularly when combined with other authentication factors. By carefully considering the design, implementation, and ongoing maintenance of OTP systems, organizations can strengthen their security posture and protect sensitive data effectively.

8.REFERENCES:

1. **RFC Documents**:

   RFC 6238: TOTP

   RFC 4226: HOTP

2. **NIST Special Publication**:

   NIST SP 800-63B

3. **Books**:

   "Authentication: From Passwords to Public Keys" by Richard E. Smith (Available on platforms like Amazon, Google Books, etc.)

4. **Academic Journals and Conference Papers**:

   You can find academic papers through databases like Google Scholar, IEEE Xplore, ACM Digital Library, etc.

5. **Security Blogs and Websites**:

   OWASP

   SANS Institute

6. **Vendor Documentation**:

   Visit the official websites of OTP solution vendors such as Google Authenticator, Authy, RSA SecurID, etc.

7. **Industry Reports and Surveys**:

   Reports from Gartner, Forrester, IDC, etc., are often available on their respective websites or through subscription services.

8. **Standards Organizations**:

   ISO, IETF, ANSI, etc., have official websites where you can access standards documents and publications.