Mastering Project Utilization: A User's Guide

Introduction:

In this project, we have developed a se **Mastering Project Utilization: A User's Guide**cure system that caters to three distinct roles: teacher, student, and admin. Each role is associated with a unique public key, ensuring secure communication within the server. The primary objective of this project is to enhance the security of student copies and marks, providing a safe and controlled environment for educational interactions.

Scope of the Project:

We have implemented the project using the Laravel framework, utilizing additional tools such as Forge, jsencrypt, and crypto-js. The main focus of the project is to secure student copies and their corresponding marks, while also managing user roles and access rights.

Methodology:

For the client-side implementation, we utilized JavaScript to handle user interface interactions, thereby enhancing the user experience while maintaining robust security measures. On the server side, we employed PHP to manage data processing and interactions with the database.

A noteworthy security feature we integrated involves the use of session keys. Specifically, for each assignment, a unique session key is generated. This session key is then employed to encrypt the assignment. To ensure dual-layer security, the session key is encrypted with both the corresponding teacher's public key and the student's public key associated with the assignment's owner.

On one end, the session key is encrypted with the teacher's public key and stored in the database. On the other end, it is also encrypted with the student's public key. By taking this approach, we create a system where both the teacher and the respective student can decrypt the uploaded assignment. They achieve this decryption by employing the session key encrypted with their respective public keys and then utilizing their private keys for decryption.

User Roles and Access:

Upon logging into the system, users are initially assigned the "student" role. If a user desires a different role, they must submit a request to the admin. The admin is responsible for reviewing and approving these requests, ensuring that role changes are legitimate and authorized.

Security Measures Implemented:

Public Key Infrastructure: Each role (teacher, student, admin) is associated with a unique public key, enhancing data encryption and security during communication.

Data Encryption: Sensitive data, such as student copies and marks, are encrypted using jsencrypt and crypto-js libraries to prevent unauthorized access.

Role Approval Workflow: The admin-controlled role change system adds an extra layer of security by requiring admin approval for any role changes.

Conclusion:

We implement role-based access control, encryption, and a streamlined request and approval system, we have created a secure and efficient platform for educational interactions.

Future Perspectives:

In the future, while we were unable to accomplish everything we initially intended, there is still room for enhancing and refining the project.