MINI PROJECT REPORT

On

**Secure Text Transfer Using Diffie-Hellman**

**Key Exchange Based On Cloud**

Submitted in partial fulfilment of the requirement

Of

University of Mumbai

For the Course

Cloud Computing

in

Computer Engineering (SEM VIII)

Submitted By:

**Siddharth Byale (117A1009)**

**Unnati Choudhari (117A1013)**

**Shardul Doke (117A1018)**

Subject Incharge:

**Prof. Masoodha Modak**



Department Of Computer Engineering

Sies Graduate School of Technology

UNIVERSITY OF MUMBAI

Academic Year 2020 – 2021

**CERTIFICATE**

This is to certify that the requirements for the project report entitled ‘**Secure Text Transfer Using Diffie-Hellman Key Exchange Based On Cloud**’ has been successfully completed by the following students:

Name Roll No

Siddharth Byale (117A1009)

Unnati Choudhari (117A1013)

Shardul Doke (117A1018)

in partial fulfilment of the course Cloud Computing in Computer Engineering (VIII SEM) of Mumbai University in the Department of Computer Engineering, SIES graduate school of technology, Nerul– 400706 during the Academic Year 2020-21.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Prof. Masoodha Modak**

**Subject Incharge**

Department of Computer Engineering

SIES graduate school of technology

Nerul– 400706

**PROJECT APPROVAL**

This project entitled “**Secure Text Transfer Using Diffie-Hellman Key Exchange Based On Cloud**” by Siddharth Byale,Unnati Choudhari and Shardul Doke are approved for the course Cloud Computing in Computer Engineering (VIII sem) of Mumbai University in the Department of Computer Engineering.

Examiners:

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Subject Incharge:

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date:

Place:

Department of Computer Engineering

SIES graduate school of technology

Nerul– 400706

**DECLARATION**

We declare that this written submission for the Cloud Computing mini project entitled “**Secure Text Transfer Using Diffie-Hellman Key Exchange Based On Cloud**” represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any ideas / data / fact / source in our submission. We understand that any violation of the above will cause disciplinary action by the institute and also evoke penal action from the sources which have not been properly cited or from whom prior permission has not been taken when needed.

Project Group Members:

Siddharth Byale: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Unnati Choudhari: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Shardul Doke: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date:

Place:

Department of Computer Engineering

SIES graduate school of technology

Nerul– 400706**Acknowledgement**

We would like to express our gratitude to our Cloud Computing teacher Prof. Masoodha Modak for her guidance and support in completing our project.

We would also like to extend our gratitude to our principal Prof. Atul Kemkar and HOD Ma’am Aparna Bannore for providing us with all the facilities that were required to complete our project.

**Project Team**

Siddharth Byale (117A1009)

Unnati Choudhari (117A1013)

Shardul Doke (117A1018)

**Abstract**

Advent of Cloud Computing has been a phenomenal phase in the history of computer science. It provided capabilities to solve many problems that were earlier deemed impossible to be computed by a machine. It removed the pressure from those responsible for manufacturing better machines to keep up with the increasing complexity of the problems that the machines are intended to solve. Cloud Computing provided a platform for better utilization of the resource spread across the world. Being a nascent field, it is crowded with many different problems that the engineers and scientists are working assiduously to eliminate. One of the main drawbacks with cloud is security. So, this project proposes a mechanism for secure file storage cloud using encryption and Diffie Hellman. The algorithm involves encrypting the file stored on the cloud and using Diffie Hellman for authenticating the user to decrypt the required file.

**Table of Contents**

| **INDEX NO.** | **CONTENT** | **PAGE NO.** |
| --- | --- | --- |
| **1.1** | Fundamentals | **1** |
| **1.2** | Objectives | **1** |
| **1.3** | Organisation of report | **1** |
| **2.1** | Literature review | **2** |
| **2.2** | Literature summary | **2** |
| **3.1.1** | Diffie Hellman | **3** |
| **3.1.2** | Prime Number | **4** |
| **3.2** | Method | **4** |
| **3.2.1** | Encryption | **5** |
| **3.2.2** | Advanced Encryption Standard(AES) | **5** |
| **3.2.3** | Requirements | **6** |
| **4.1** | Planning and Analysis | **7** |
| **4.2** | Implementation of DH | **7** |
| **4.3** | Implementation of AES | **8** |
| **4.4** | Web-Application | **8** |
| **5.1** | Summary | **13** |
| **5.2** | Future Scope | **13** |
|  | References | **14** |

**Chapter 1**

**Introduction**

**1.1 Fundamentals**

Cloud security is one of the main concerns in the cloud computing domain. Storing personal and sensitive information on a third-party storage medium poses serious risks of data theft and data misuse by any person with malicious intent. The threat is so humongous that it has dissuaded governments and many other big organizations from migrating their operations on a cloud platform. The traditional methods of securing files and information are superfluous in the scenario of cloud. Extensive research and study is undergoing in this field to make the cloud more secure and reliable. Among these behemoth instances of research, some of the methods that stand out include AES encryption and Diffie Hellman Key Exchange. The latter method is so powerful that it may take millions of years for even the most powerful computers of current times to crack the code and read the file. Our approach proposes a method that involves encrypting the file using any standard encryption technique and using Diffie Hellman for user authentication. In this way the files can be saved in a public domain securely without the threat of being used by any unauthorised person.

**1.2 Objectives**

The main objective is to double ensure that your secret message is sent secretly without outside interference of hackers or crackers

Advantage of the following:

* Fast and easy way to send secure text messages.
* Use two-way encryption technique.
* The database is stored on the cloud using AWS, hence it is more secure

**1.3 Organization of the Report**

Chapter 1 - Introduction : Why is cloud based key exchange important.

Chapter 2 - Literature Survey : Existing System for Key Exchanges and Drawbacks.

Chapter 3 - Overview : Basics of the Diffie Hellman Algorithm.

Chapter 4 - Implementation : Cloud based implementation of Diffie Hellman.

Chapter 5 - Conclusion and Future Scope: Areas for future improvement on the project.

**Chapter 2**

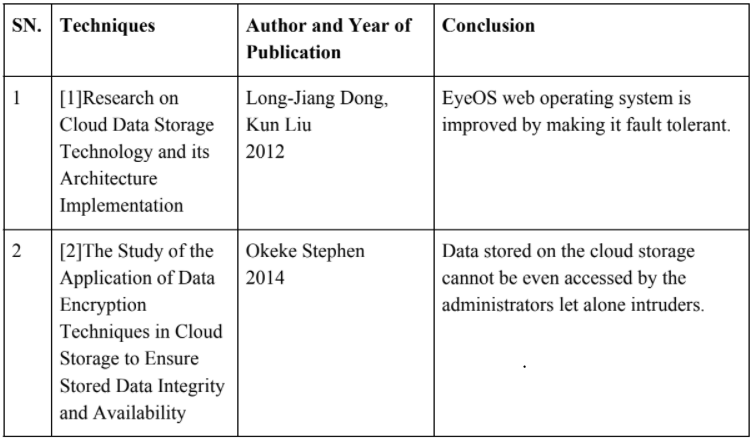
**Literature Survey**

**2.1 Literature Review**

The first paper explains the concept of cloud computing and the various cloud storage techniques like the GFS (Google File System) and HDFS (Hadoop Distributed File System). It then points out to the fact that the eyeOS web operating system which implements the cloud storage architecture is neither fault tolerant nor highly reliable. Moreover, access performance is quite low. Therefore, it can be improved by implementing the distributive feature of the HDFS and eventually achieve fault tolerance.

The second paper points out to the issues of cloud storage like malicious data theft and data destruction causing quite a loss to the government organizations and other private companies as well. Therefore, a symmetric encryption algorithm is implemented to ensure secure data storage on the cloud. When the session is logged out, this approach makes the use of the encryption key that acts as the primary authentication for the user.

**2.2 Literature summary**

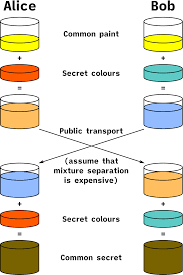


**Chapter 3**

**Overview**

**3.1.1 Diffie Hellman Key exchange**

Diffie–Hellman key exchange (DH) is a method of securely exchanging cryptographic keys over a public channel and was one of the first public-key protocols named after Whitfield Diffie and Martin Hellman. [1] DH is one of the earliest practical examples of public key exchange implemented within the field of cryptography. In public key cryptosystem, enciphering and deciphering are governed by distinct keys, E and D, such that computing D from E is computationally infeasible (e.g., requiring more than 10100 instructions). The enciphering key E can thus be publicly disclosed without compromising the deciphering key D. This was the main ideology behind the Diffie-Hellman Key Exchange Protocol. Each user of the network can, therefore, place his enciphering key in a public directory. This enables any user of the system to send a message to any other user enciphered in such a way that only the intended receiver can decipher it. As such, a public key cryptosystem is a multiple access cipher. A private conversation can therefore be held between any two individuals regardless of whether they have ever communicated before. Each one sends messages to the other enciphered in the receiver’s public enciphering key and deciphers the messages he receives using his own secret deciphering key.

****

Diffie–Hellman key exchange establishes a shared secret between two parties that can be used for secret communication for exchanging data over a public network. The above conceptual diagram illustrates the general idea of the key exchange by using colours instead of very large numbers. The process begins by having the two parties, Alice and Bob, agree on an arbitrary

starting colour that does not need to be kept secret. In this example the color is yellow. Each of them selects a secret color that they keep to themselves – in this case, orange and blue-green. The crucial part of the process is that Alice and Bob each mix their own secret color together with their mutually shared color, resulting in orange-tan and light-blue mixtures respectively, and then publicly exchange the two mixed colours. Finally, each of the two mixes the color he or she received from the partner with his or her own private color. The result is a final color mixture (yellow-brown in this case) that is identical to the partner's final color mixture. If a third party listened to the exchange, it would be computationally difficult for this party to determine the secret colors. In fact, when using large numbers rather than colors, this action is computationally expensive for modern supercomputers to do in a reasonable amount of time.

**3.1.2 Prime Number**

A prime number (or a prime) is a natural number greater than 1 that cannot be formed by multiplying two smaller natural numbers. The only user-defined pre-existing parameter in the Diffie-Hellman protocol is the selection of prime numbers. The prime number p should be large enough to defend against the known attacks against it. The most efficient attack is NFS (attack on the network file system); that has been used against numbers on the order of 2^768 (a 232-digit number). It would appear wise to pick a p that's considerably bigger than that; around 1024 bits at a minimum, and more realistically at least 1536 bits. Another property about p is that p−1 should have a large prime factor q, and one should know what the factorization of p−1 is. If we pick a random prime p, and a random generator g, well, we’re probably secure, but we won't be certain (and we might leak a few bits of the private exponent if the order of your random g happens to have some small factors).

**3.2 METHOD**

*p* is a prime number

*g* is a primitive root modulo of *p*

* Alice and Bob agree to use a modulus p = 23 and base g = 5
* Alice gets her private key (key which she should not share with anyone) generated as 4.
* Thus, public key generated for Alice shall be 5^4 %23 = 625%23 = 4
* Bob gets his private key (key which he should not share with anyone) generated as 3.
* Thus, public key generated for Bob shall be 5^3 %23 = 125%23 = 10
* Now, Alice gets the public key of Bob and generates a secret key. i.e. (public key of Bob ^ Private Key of Alice) mod p => (10^4 ) % 23 => 10000 % 23 => 18
* On the other side, Bob also uses a similar method to generate a secret key i.e. (public key of Alice ^ Private Key of Bob) mod p => (4^3 ) % 23 => 64 % 23 => 18

Thus, it is proven that mathematically, Alice and Bob generate the same key without each one of them knowing the other one’s private key. This is the implementation of Diffie-Hellman Key Exchange Protocol

**3.2.1 Encryption**

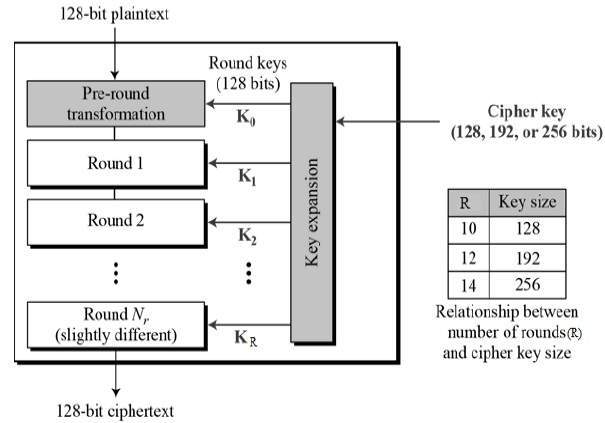
Encryption is widely used on the internet to protect user information being sent between a browser and a server, including passwords, payment information and other personal information that should be considered private. Organizations and individuals also commonly use encryption to protect sensitive data stored on computers, servers and mobile devices like phones or tablets. There are various encryption techniques that are present some of which are: Triple DES Blowfish RSA Twofish AES The technique that we have used in our project is AES and it is described below.

**3.2.2 Advanced Encryption Standard**

Criteria for being chosen as the next AES algorithm included the following:

* Security. Competing algorithms were to be judged on their ability to resist attack -- as compared to other submitted ciphers. Security strength was to be considered the most important factor in the competition.
* Cost. Intended to be released on a global, nonexclusive and royalty-free basis, the candidate algorithms were to be evaluated on computational and memory efficiency.
* Implementation. Factors to be considered included the algorithm's flexibility, suitability for hardware or software implementation, and overall simplicity.

The more popular and widely adopted symmetric encryption algorithm nowadays is the Advanced Encryption Standard (AES). It is found to be at least six times faster than triple DES. A replacement for DES was needed as its key size was too small. With increasing computing power, it was considered vulnerable against exhaustive key search attacks. Triple DES was designed to overcome this drawback, but it was found to be slow. The AES has three fixed 128-bit block ciphers with cryptographic key sizes of 128, 192 and 256 bits. Key size is unlimited, whereas the block size maximum is 256 bits. The AES design is based on a substitution-permutation network (SPN) and does not use the Data Encryption Standard (DES) Feistel network. The diagram below shows the implementation of AES encryption technique.



## 

## **3.2.3 REQUIREMENTS**

* Python
* Python flask
* Python crypto
* Amazon Web Services(AWS)
* Elastic Compute Cloud(EC2)

**Chapter 4**

**Implementation**

**4.1 Planning and analysis**

The main task of this project was to provide as secure a file storage on the cloud as possible. So, several issues had to be sorted out like :

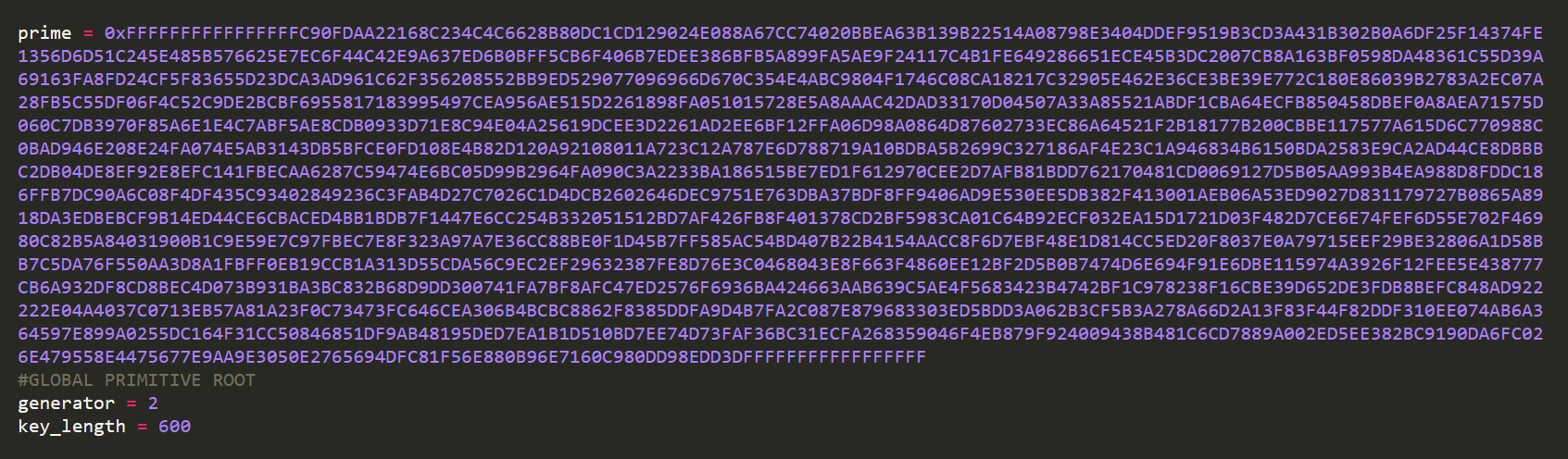
* Where the file needs to be encrypted.?
* How must the user be authenticated?
* What will be the key for AES encryption?
* How will this key be related to the Diffie Hellman Key exchange protocol? We first thought of encrypting the text online, but the attack man in the middle made us not choose that method. We also learnt about an attack abbreviated as NFS. NFS in an attack in which a key of order 2768 could be computed. This was approximately 232 digits. Thus, we came to a conclusion that a larger prime number is needed in the process. Therefore, we used a prime number having 600 digits. While analysing all these questions we came upon this course of action. To provide greater control to the owner of the file we encrypted the file on the owner’s computer itself using an application. The Diffie Hellman was used to generate the file and only those users who have the final same key from this process would be able to decrypt the file. The final key generated from Diffie Hellman, being the same for both intended participants, was used as the basis for the key for AES encryption. The following subsections describe the different parts of the project in greater detail.

**4.2 Implementation of Diffie Hellman**

This section discussed the implementation of the Diffie-Hellman algorithm.. This module had four methods to execute. The tasks were:

1. Generate a private key of given length for a new user
2. Generate a public key for a user using his private key
3. Generate secret key based on a given public and private key

First, we needed a prime number. Thus, we hard coded the prime number



This prime number is of length 600, which is much bigger and safe against NFS attacks. It can incorporate many users, i.e. a big number of users could be assigned a unique key less than this prime number. We also found out that one of the primitive roots of this prime number is 2, thus we hard coded this and used it in carrying out all the above function

**4.3 Implementation of Encryption**

This section discusses the implementation of the encryption algorithm used for securing the text. We used the PyCrypto library which already had the implementation of the AES algorithm.. AES uses base 64 encoding and encodes the text in UTF-8 format. It adds some extra bits used as padding bits to make the text more secure

**4.4 Web Application**

Let us take our previous example of user p and user q and lets try sending a text file from user p to user q using our Application.

First user p needs to register as he wants to send a file securely. He must enter a username and his name to create a user.

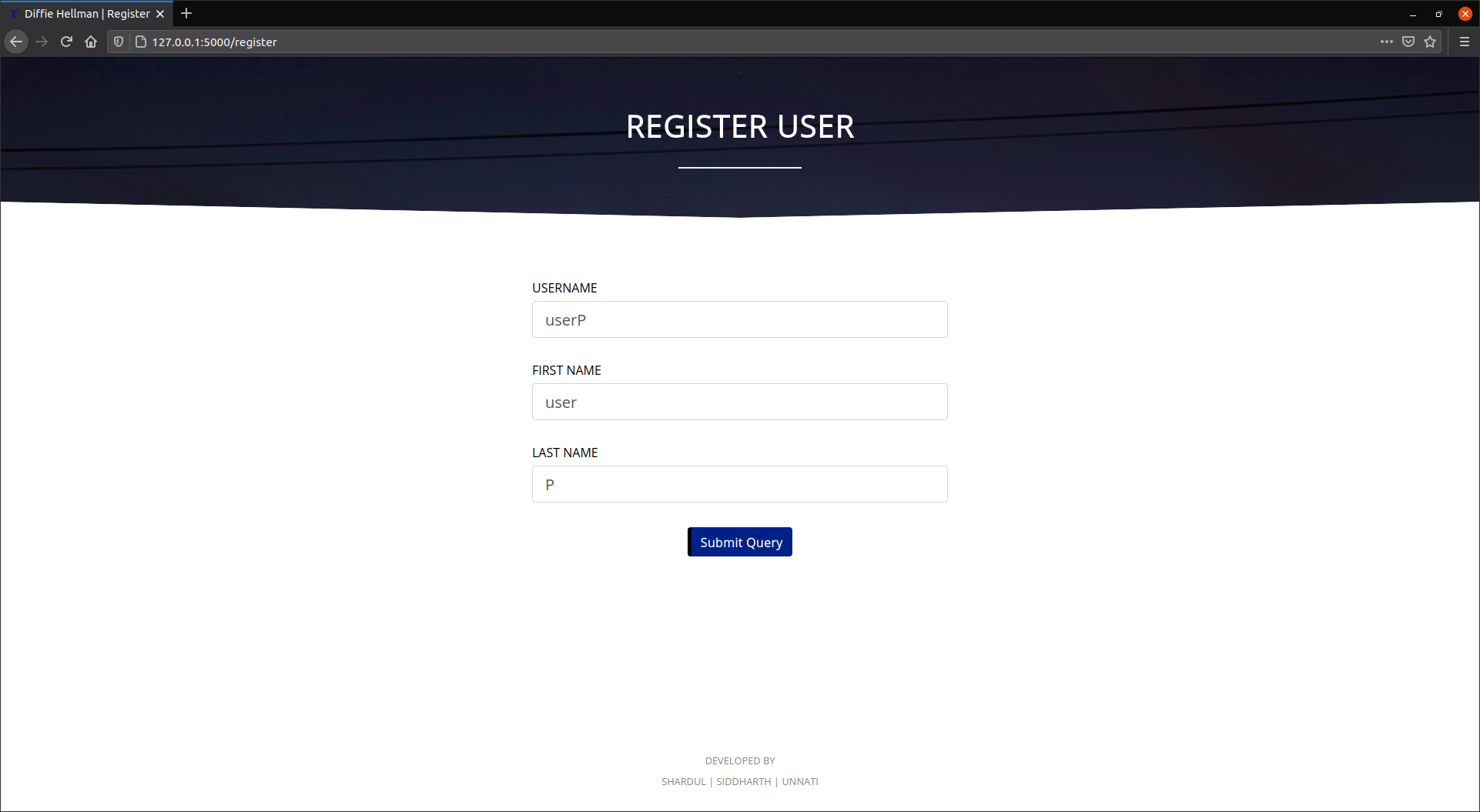
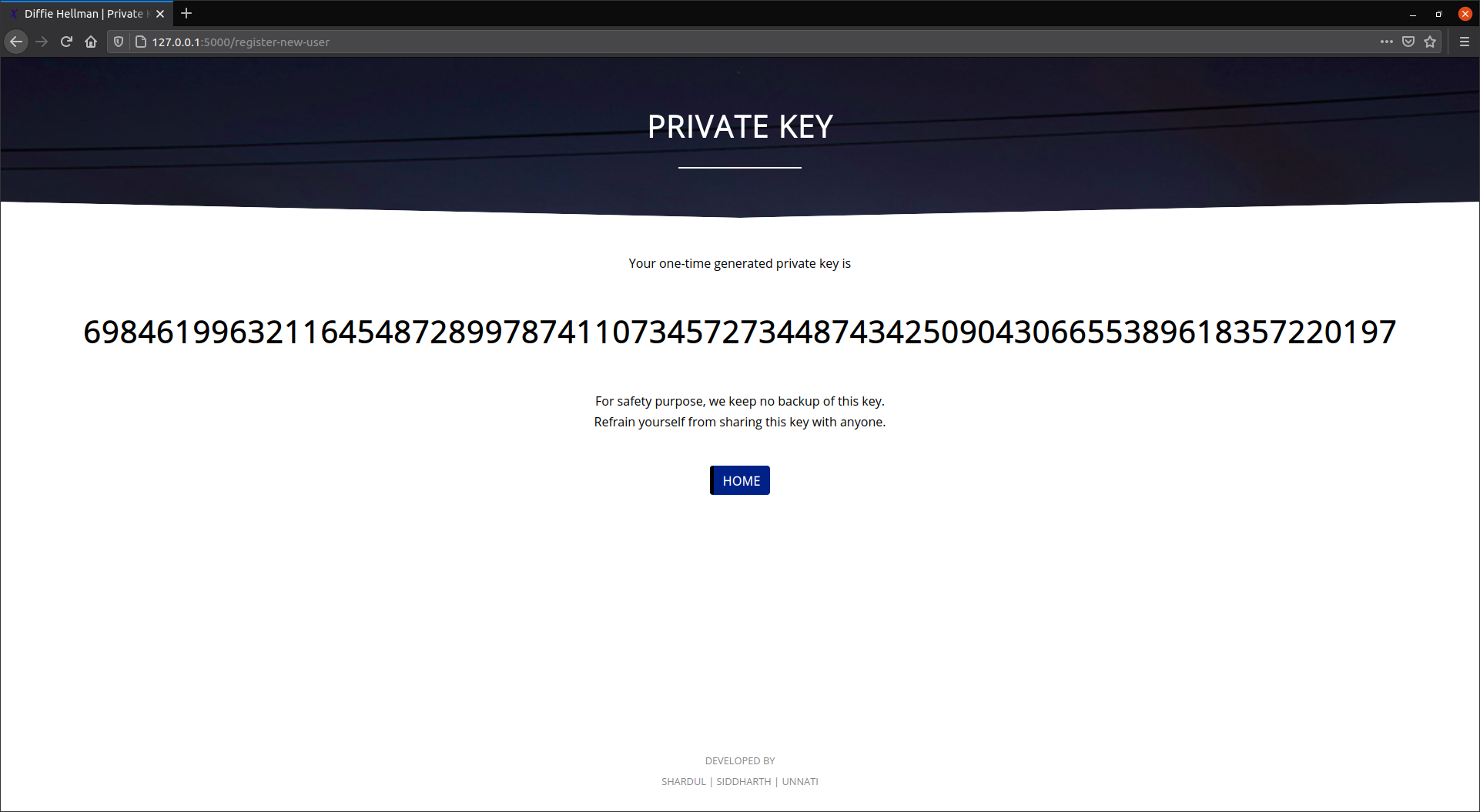
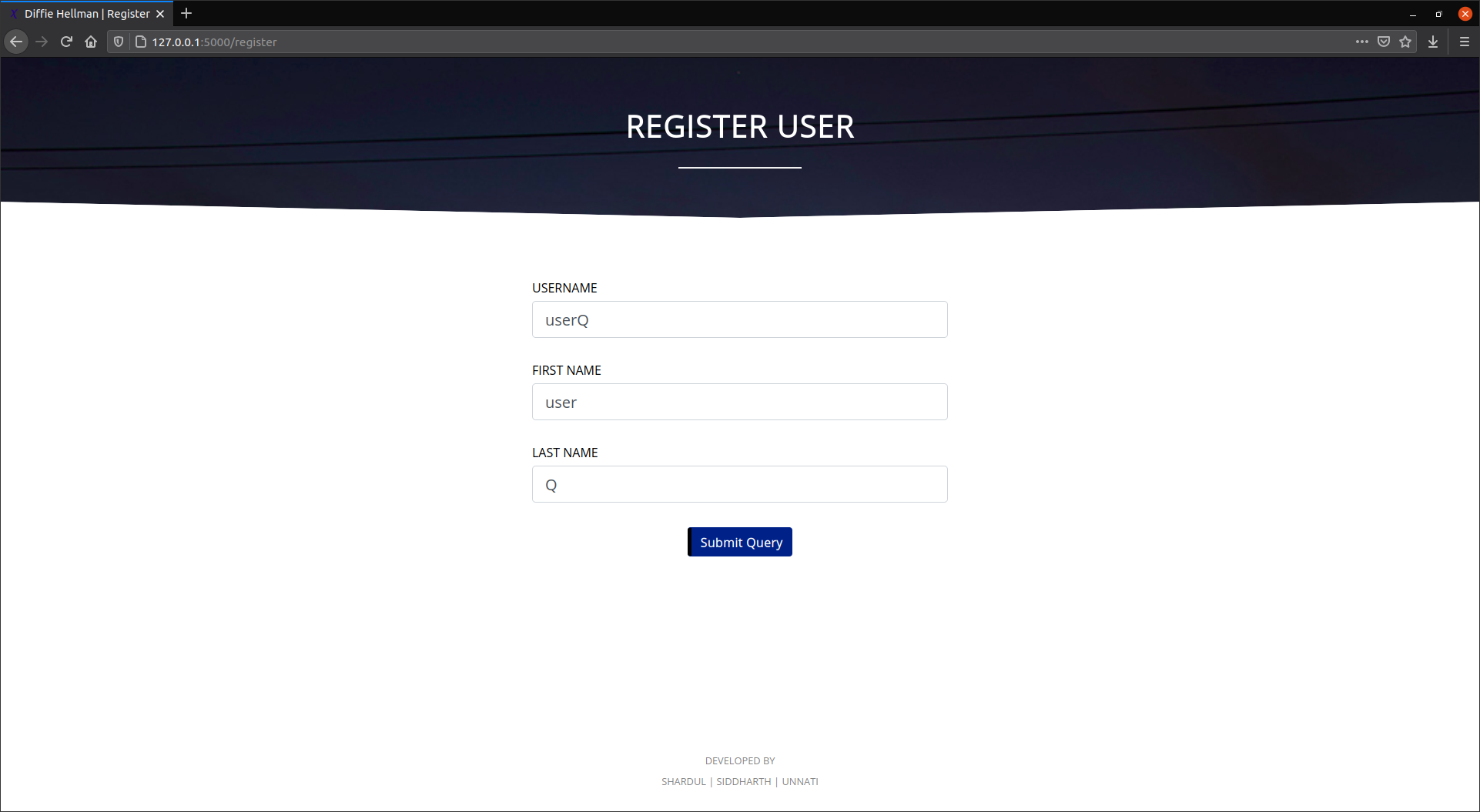
****

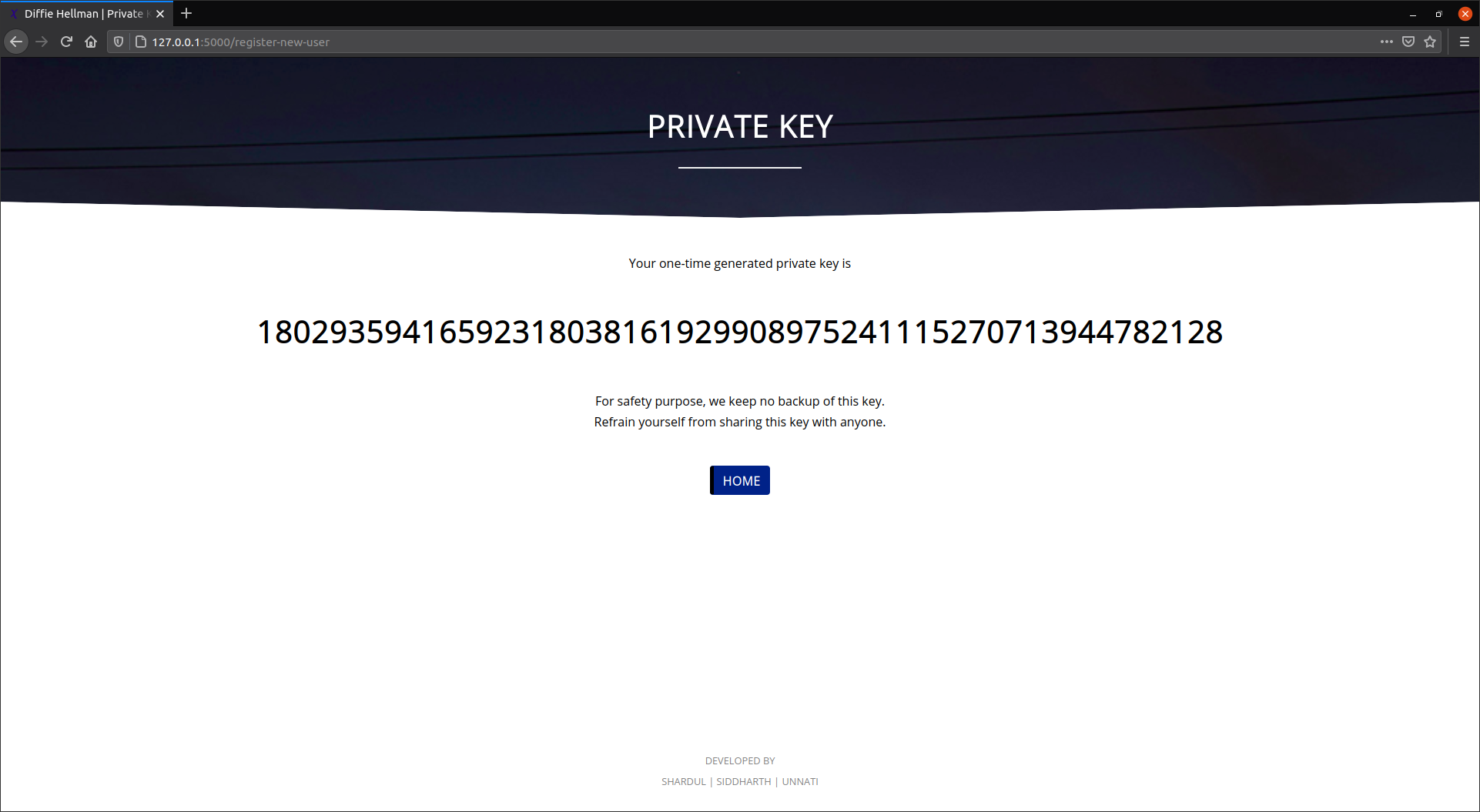
Fig.no.4 Registration of the user

As soon as userP registers himself, he will be given a private key which will be unique to userP. User p must store this key himself as this is his private key and only he should have access to this key. No copy of the private keys are stored by our application.

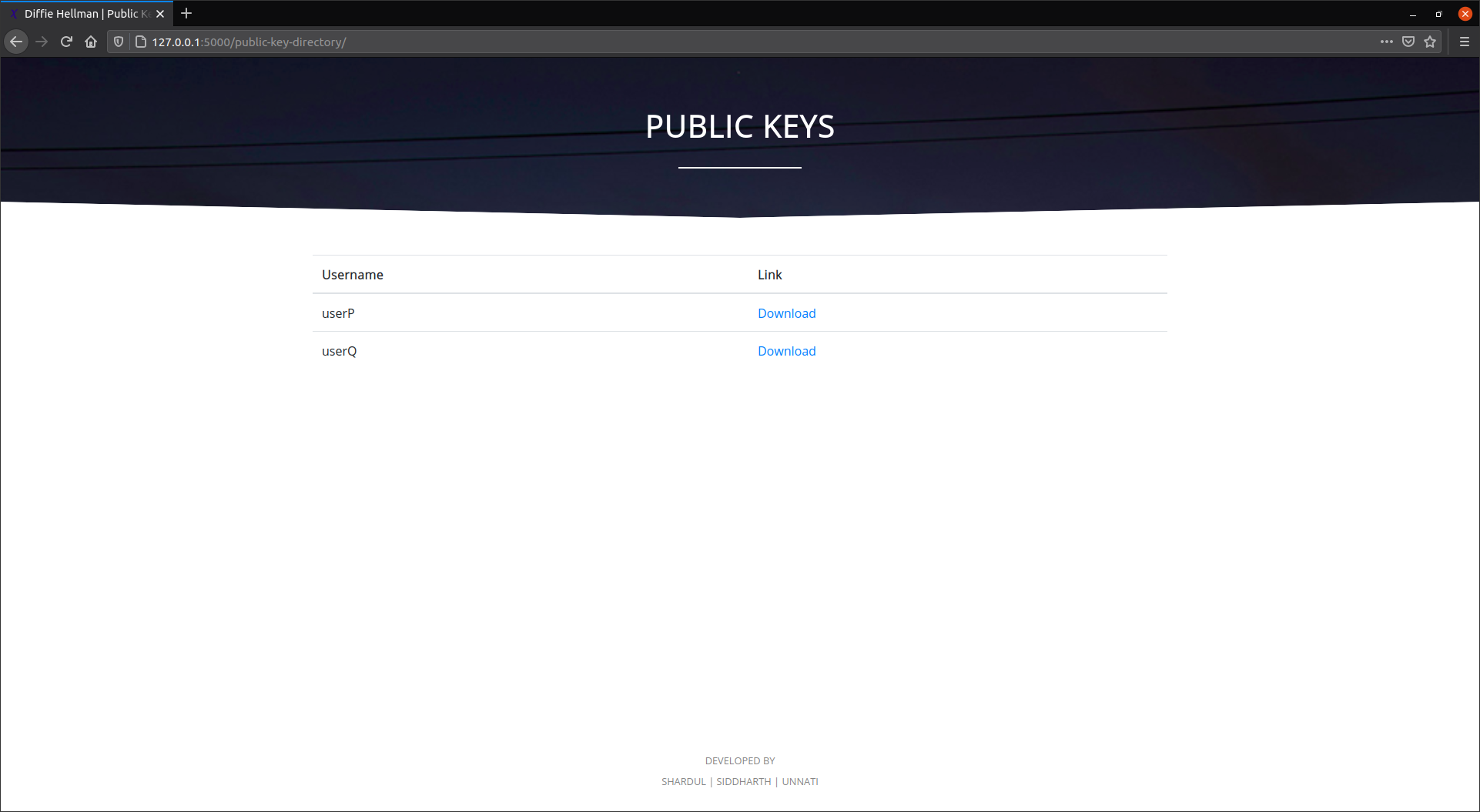
****

Let us assume that user q has also registered himself as a user and got his unique private key and stored it on his part.

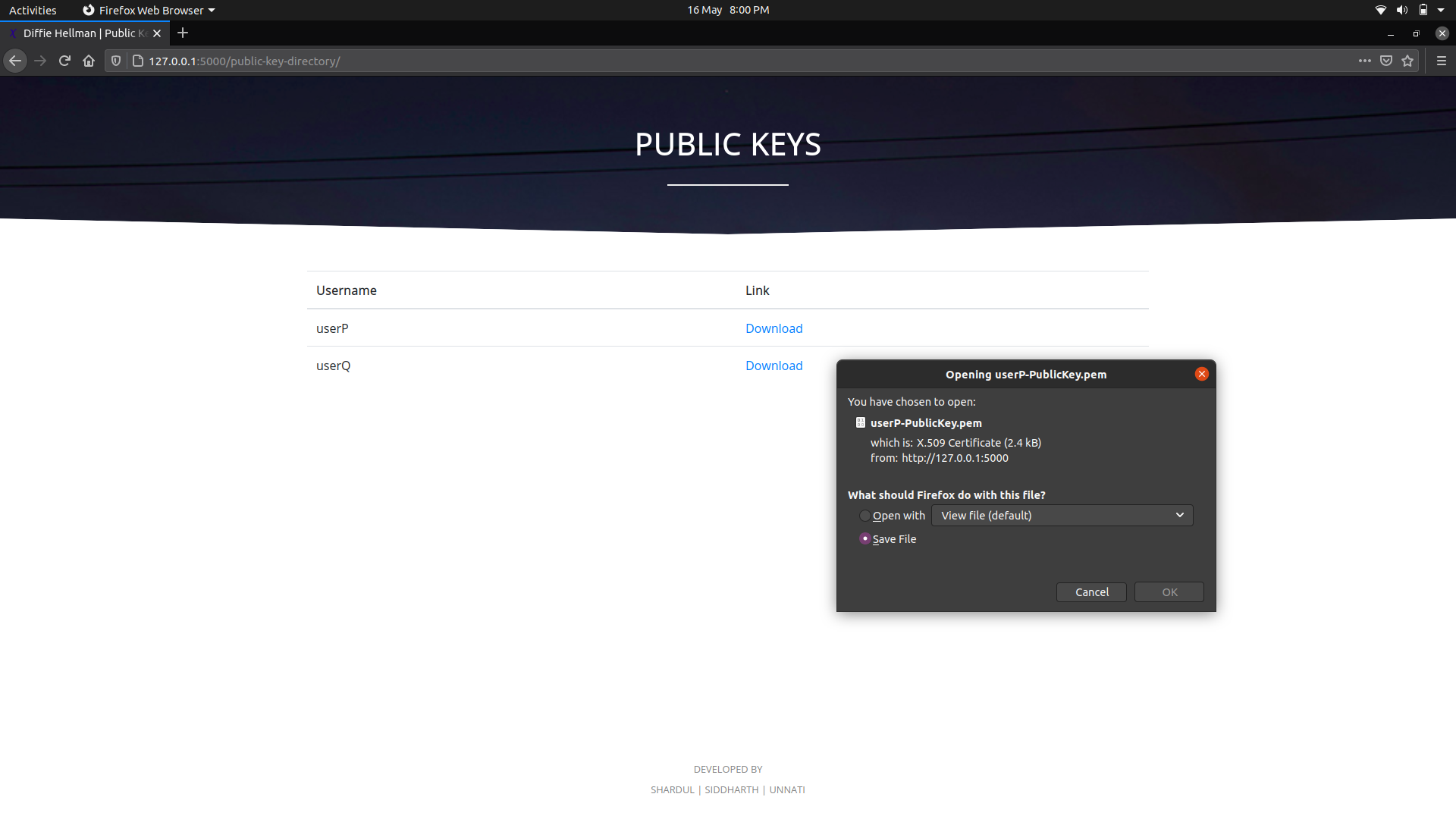
****

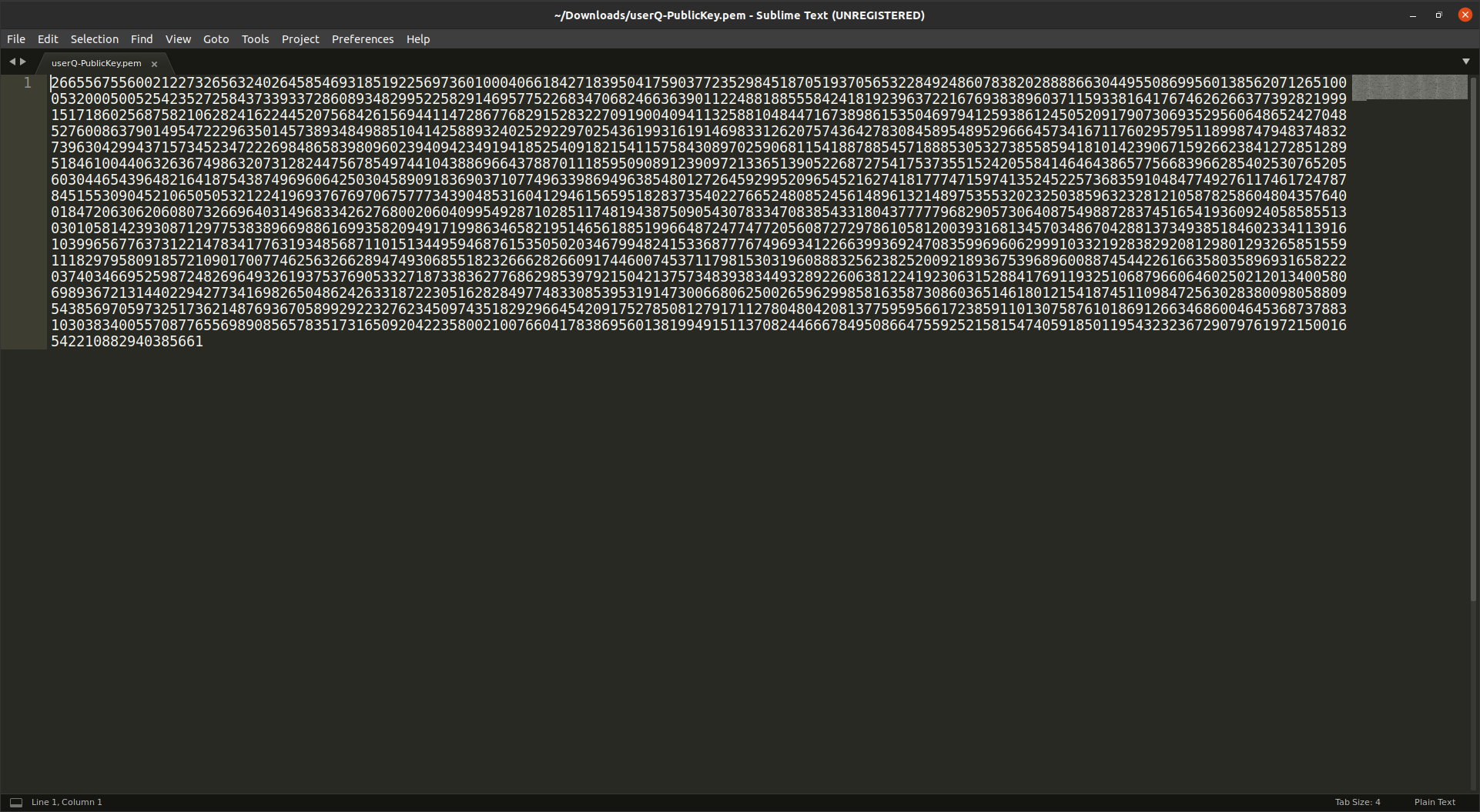
****

When a user registers himself his public key is generated automatically and stored by the system for other users to access. Anyone can get any user's public key as it will be required for the sender to encrypt his file.

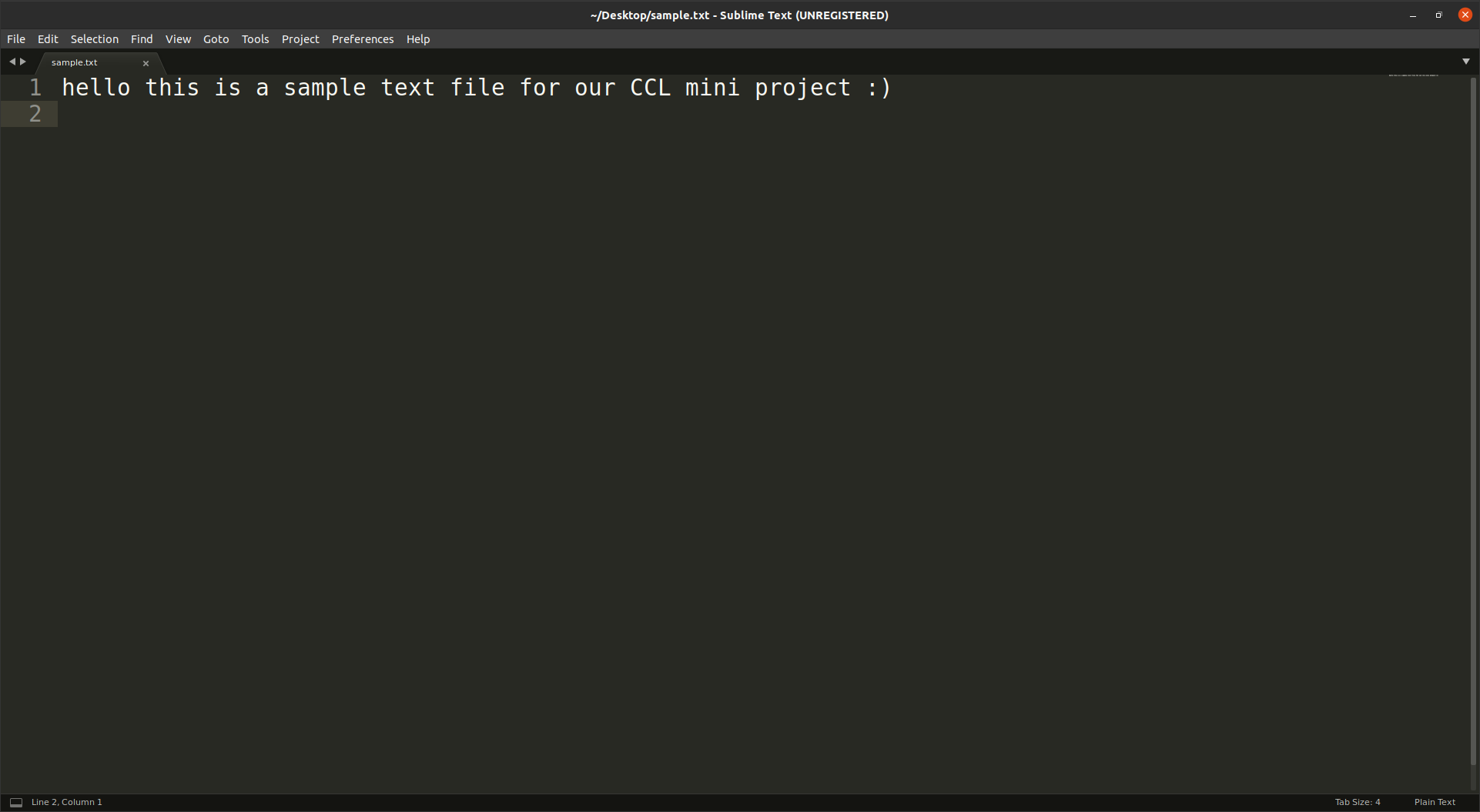
****

Now, user p will download user q’s public key as he wants to send a file to user q.

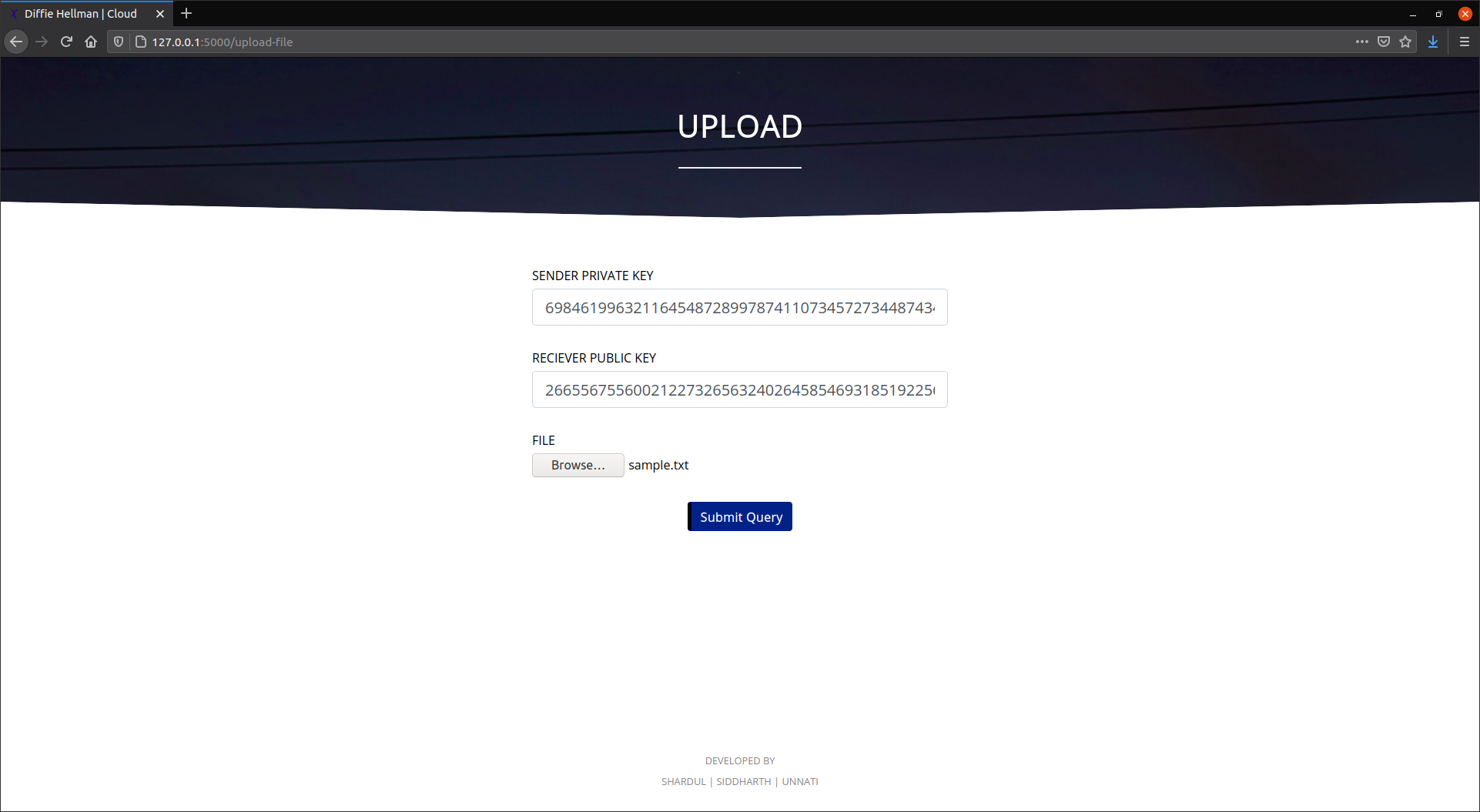
****

****

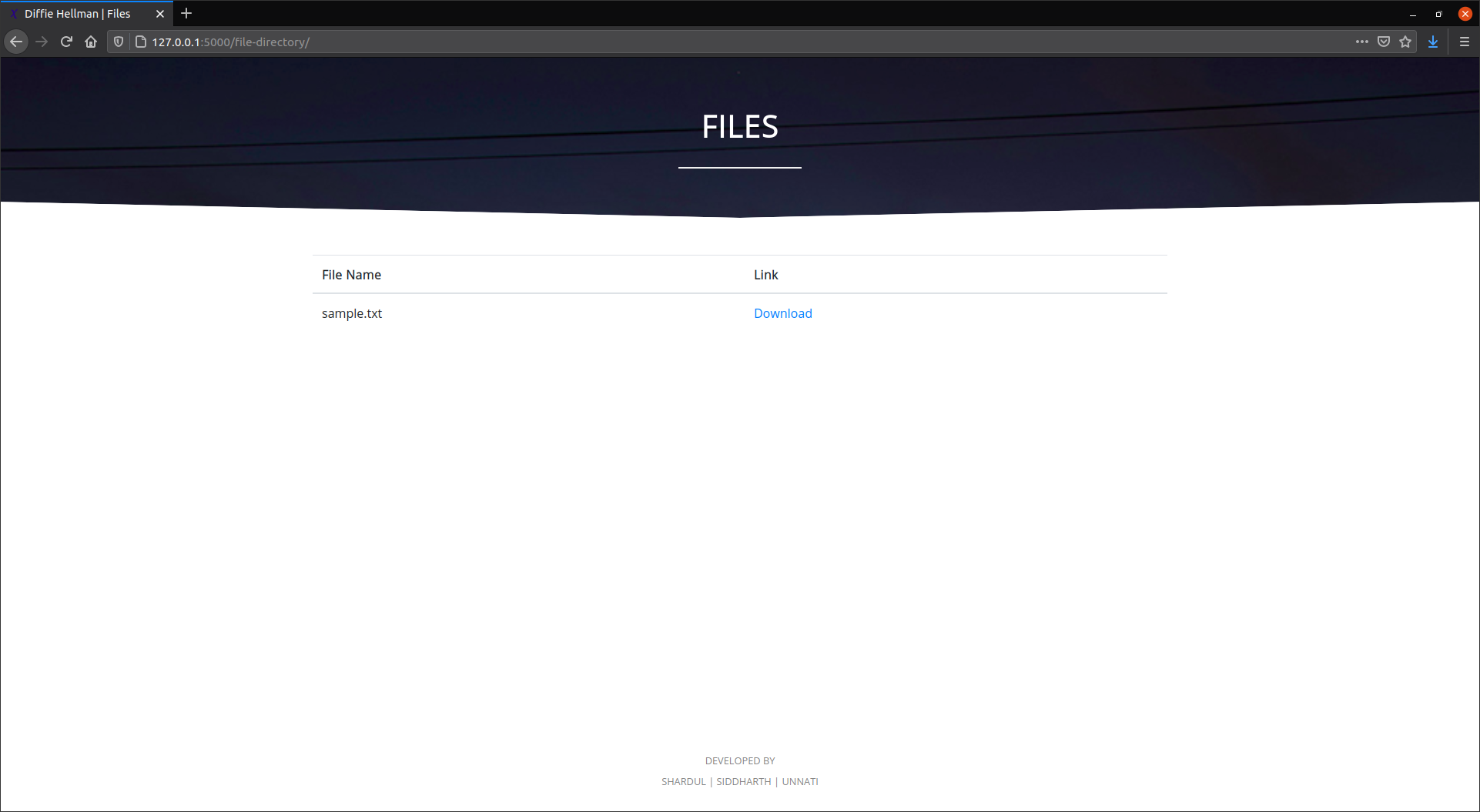
Let us say that user p wants to upload the following file using our application.

****

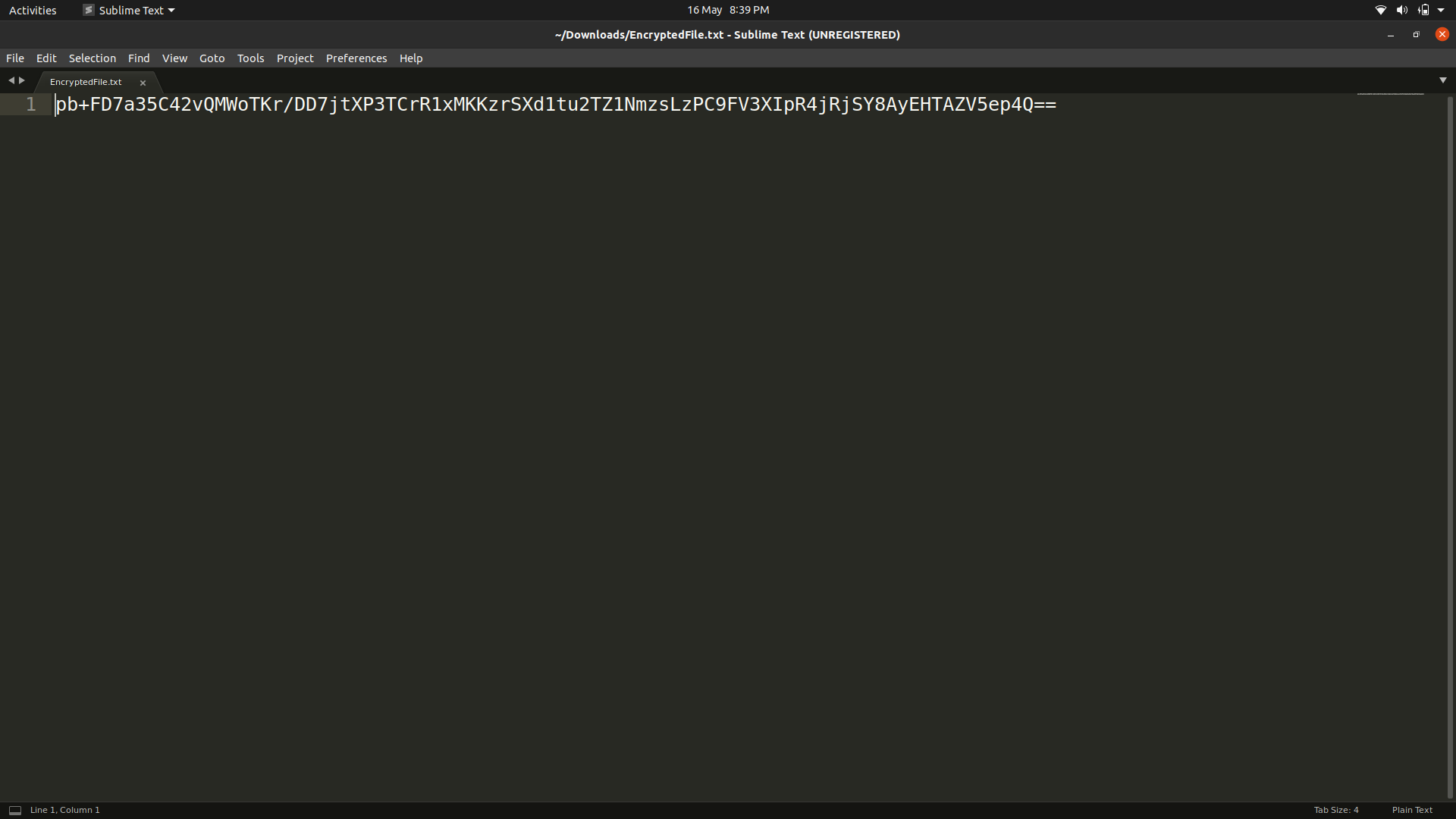
User p will input his own private key as sender and user q’s public key that he downloaded earlier as receiver. He will browse the text file from his computer and upload it using the Upload Section.

****

When any user uploads a file, the file is encrypted and stored in the system. No copy of the original file is saved for security reasons. These encrypted files can be viewed in the Files Section.

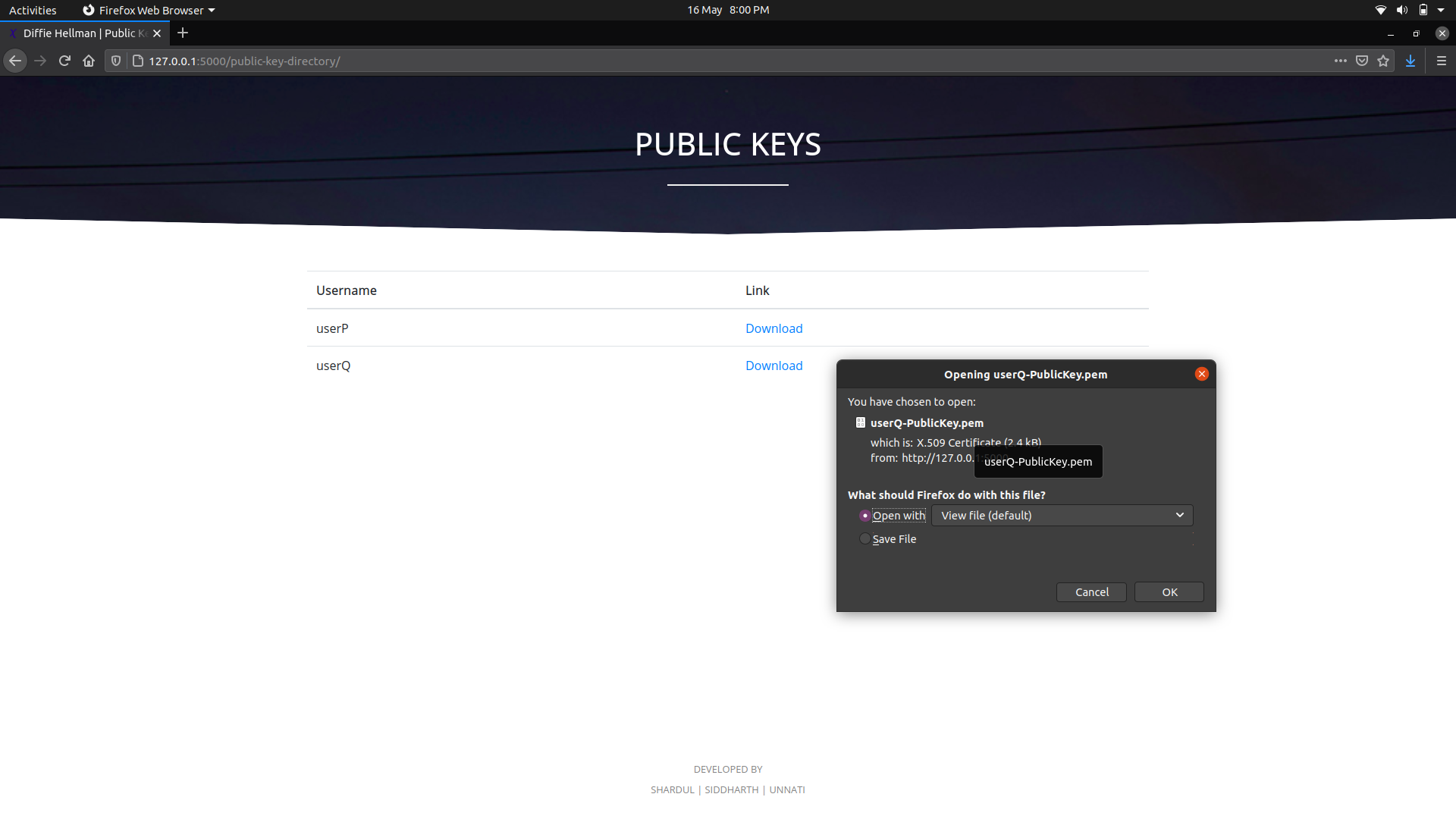
****

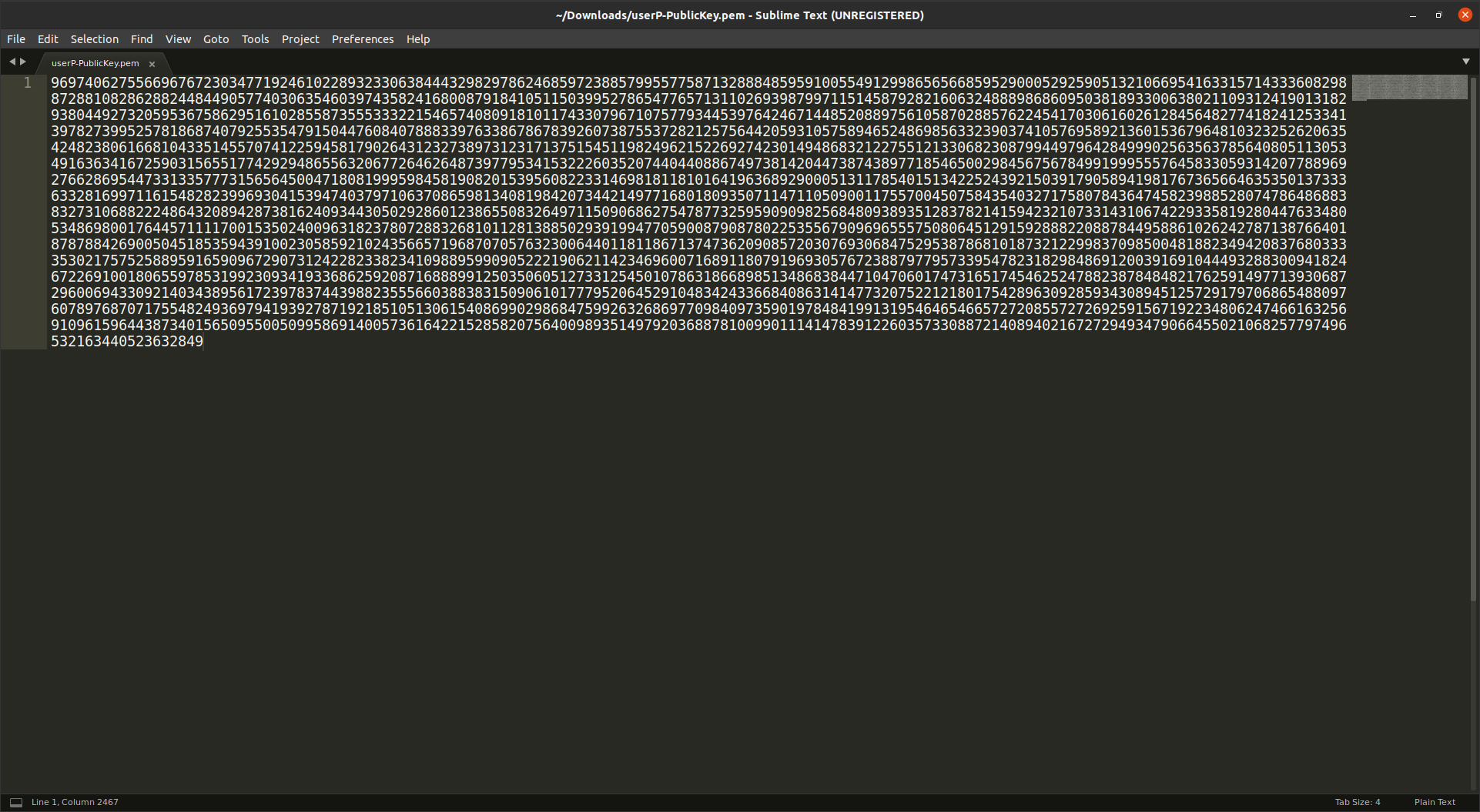
Downloading the file from here will download the encrypted copy that is stored on the application server. To Download the decrypted file user must use the Download section.

****

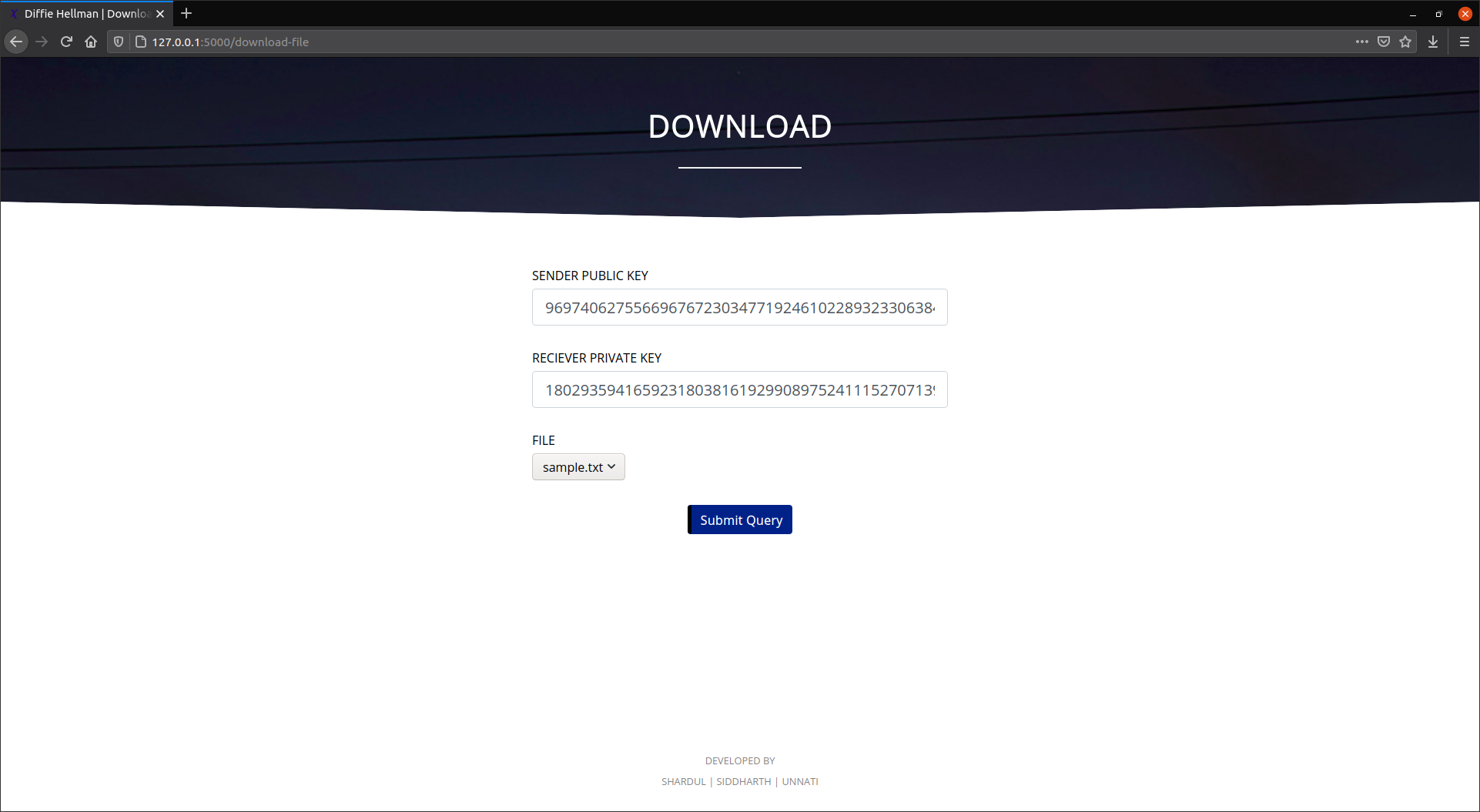
Let us come to user q and now user q wants to download the file that user p has sent to him. User q has his own private key stored by himself but needs user p’s public key to download the file.

User q will download user p’s public key from the Public Keys section.

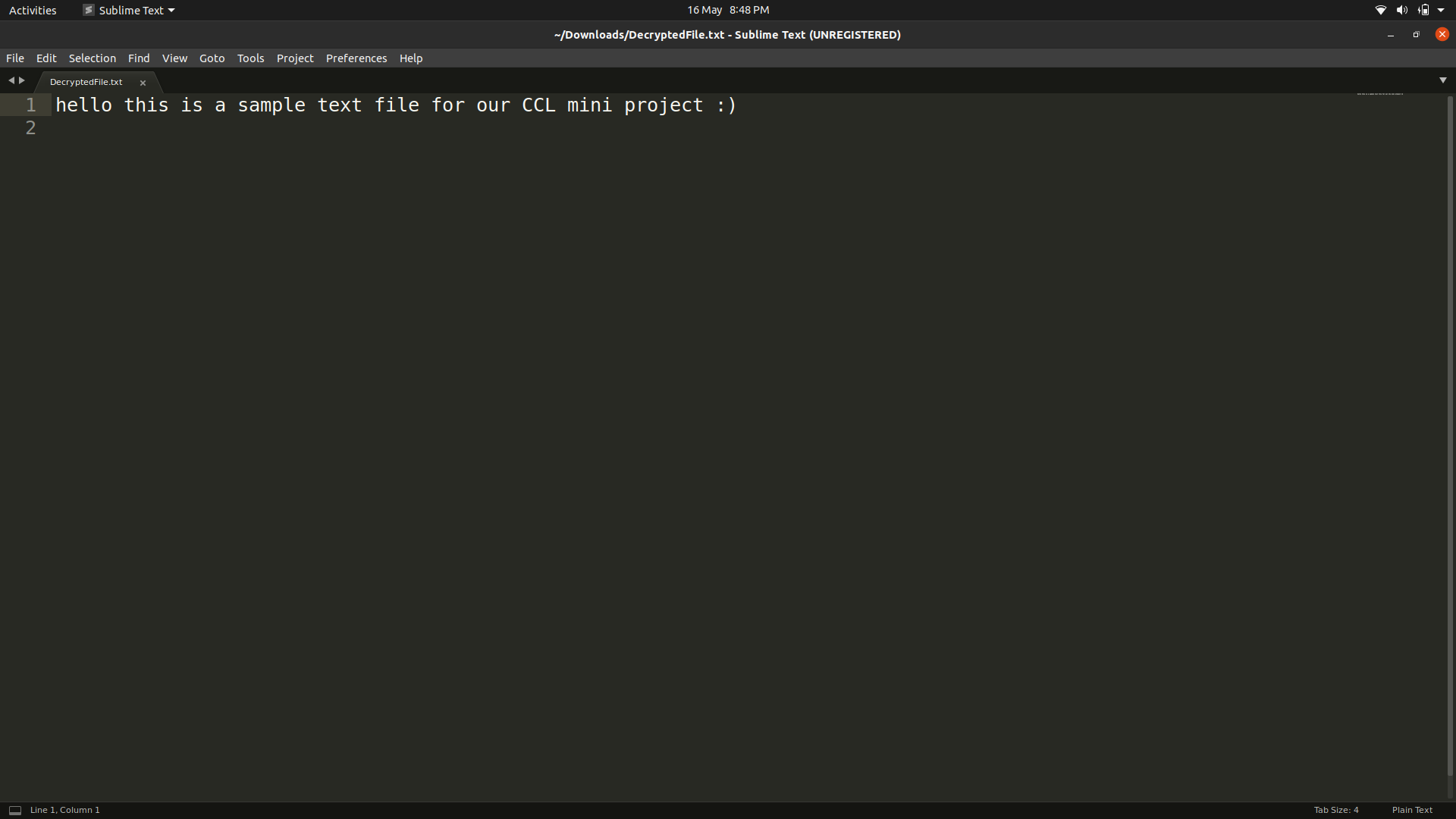
****

****

User q will input his own private key as receiver and user p’s public key that he downloaded earlier as the sender. Select the file that he wants from the dropdown list and download it.

****

In the downloaded file the message from the sender will appear in the decrypted form if the receiver has input the proper keys for decryption.

****

**Chapter 5**

**Conclusion and Future Scope**

**5.1Conclusion**

The proposed project aims to address the problem of secure file storage on the cloud. This method is a basic implementation of the proposed methodology that can be improvised and customised according to the needs. It proposes to use encryption and Diffie Hellman to provide a double layer of security to the files that are stored on the cloud.

**5.2 Future Scope**

* This is a singlecast system ,it can be modified as a multicast system so that it'll be able to send to more than one user at the same time.
* Our system doesn't have a login module,a Login module can be added so that whenever a known user logins the system will automatically fill its private key.
* The security of the current system can be greatly improved by deploying better firewalls that can protect the databases in a better capacity. In the future, the encryption standards can be upgraded to a hybrid encryption system that will better protect the data. Due to encryption the data can be guaranteed in the context of privacy; However, the fallacies around the protection provided by the cloud platform may introduce vulnerabilities that might allow an individual to damage the
* Integrity of the data. To further protect the integrity of the data, the developer can switch to a more secure platform or the system can be designed in a way that there are no vulnerabilities that might need protecting. Finally, the same system can be used in a broader perspective in an organization to exchange confidential and financial files that need end to end protection.

**References**

1. Diffie, W.; Hellman, M. (1976). "New directions in cryptography" (PDF). IEEE Transactions on Information Theory. 22 (6): 644–654. doi:10.1109/TIT.1976.1055638. Archived (PDF) from the original on 2014-11-29.

2. Kuhlman, Dave. "A Python Book: Beginning Python, Advanced Python, and Python Exercises". Archived from the original on 23 June 2012.

3. "About Python". Python Software Foundation. Retrieved 24 April 2012., second section "Fans of Python use the phrase "batteries included" to describe the standard library, which covers everything from asynchronous processing to zip files."