**Principles of Information Assurance**

**ITS – 55100**

**Final Project Report**

**Haseeb Ahmed Syed**

1. **Introduction**

This project demonstrates the process of creating a platform for secure web connection (https enabled). Using HTTPs will enable client and server interactions to be secure. The goal is to enable the client to start network connection with the distant server and give said server the authority to encrypt data. This project's objective is to create a client-server architecture that uses SSL or secure socket layer programming to ensure secure data flow among the client and server modules.

SQL injection countermeasures are another thing that has been implemented in the system to strengthen the security safeguards. SQL injection is a widely used vulnerability that allows malicious SQL queries to be injected into a database to modify it. The project uses prepared statements and extensive input validation in the server-side code to reduce this risk. To better protect against SQL injection attacks, the project is also investigating the use of prepared statements and stored procedures in database interactions. By strengthening the system's defences against any security threats and guaranteeing a secure and encrypted connection between the server—hosted on Apache—and the client website, these steps will increase the system's overall robustness.

1. **Prerequisites**

To successfully run this project we need to first install a few software. Setup for all the software are explained in depth in respective appendixes. The various software we require are:

1. Ampps.
2. OpenSSL
3. Root certificate (HTTPS)
   1. **Setup**

Please follow the steps below to setup all the modules required for accurate implementation of this project.

* + 1. **Ampps**

To install and configure AMPPS on the server, download the latest version from the official website. Follow the installation wizard to set up Apache and MySQL modules. Configure Apache's virtual hosts and ensure MySQL and PHP settings align with project requirements. Once the installation is complete, start both Apache and MySQL modules.

The Ampps pop-up task viewer should look like this

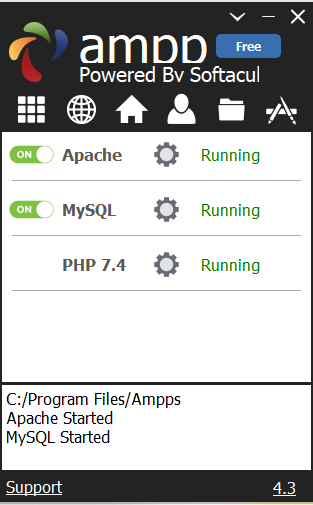


Fig 1. Ampps pop-up taskviewer

* + - 1. **Database module (MySQL)**

Once Ampps modules are up and running we can access the mysql admin page to create the databases. To access this page we can just type in the url field of our browser <https://localhost/phpmyadmin>.

The page we get looks like this

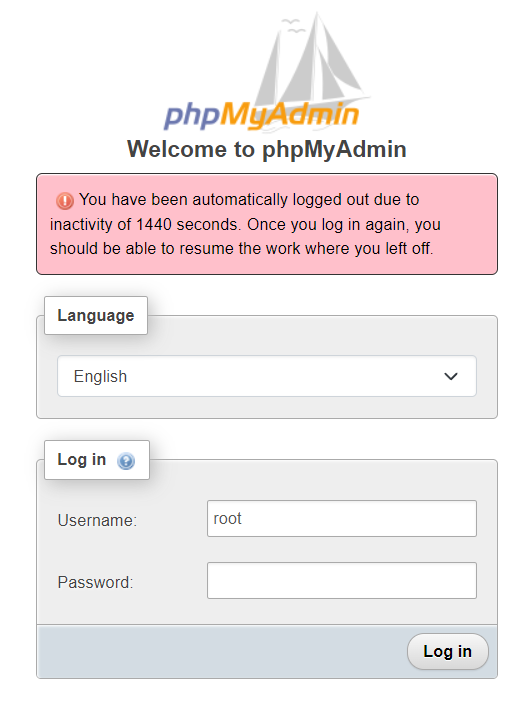


Fig 2. PhpMyAdmin landing page

Type “root” in username field and “mysql” in password field to gain access.

You will be redirected to a new page that looks like this

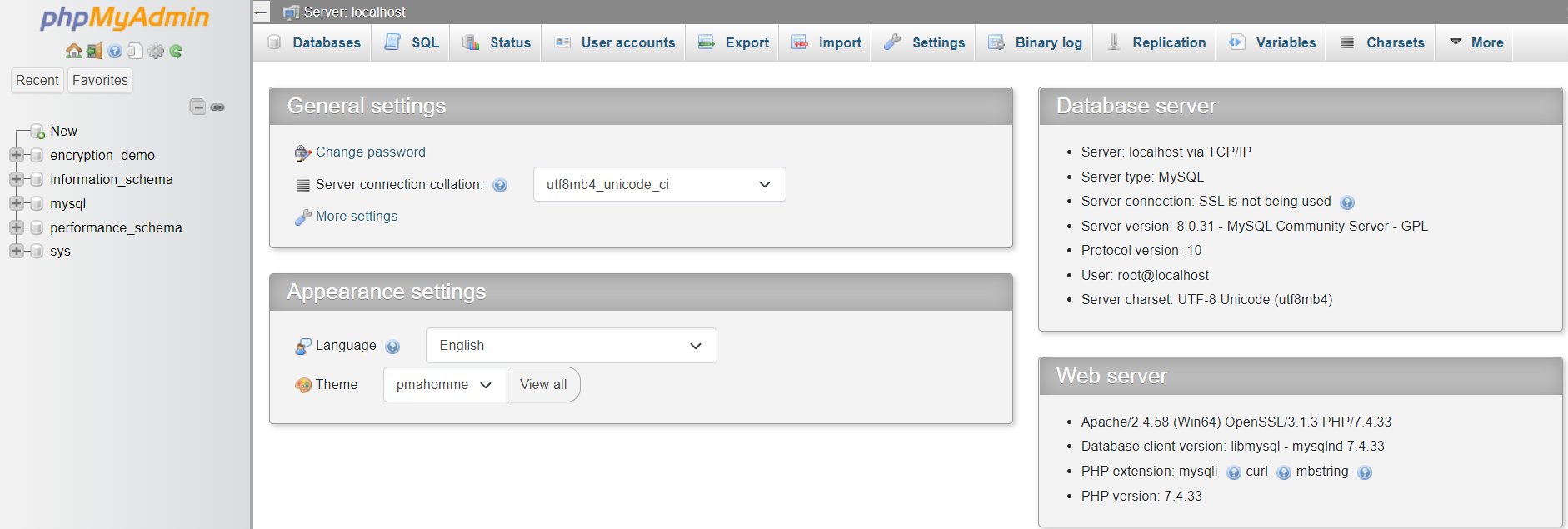


Fig 3. PhpMyAdmin homepage

You have now successfully accessed the database module of this project.

* + - 1. **Client module (Apache)**

To access the client module click on the following link: <https://localhost/IAPROJECT/LoginPage.php>

Once you get access you will land on a page like this

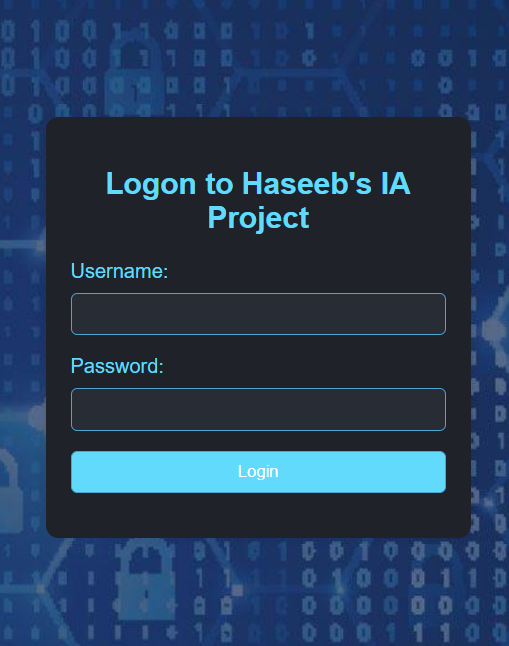


Fig 4. Login Page (Haseeb IA Project)

You have successfully gained access to client module of this project.

* + 1. **HTTPS certificate**

To install the https certificate, please download the certificate from miscellaneous folder of project. Once the certificate is downloaded, please follow the steps in provided in the videoguide to successfully activate the certificate.

Once the certificate is successfully activated you will see a lock icon on the LoginPage. Refer to the screenshots attached below

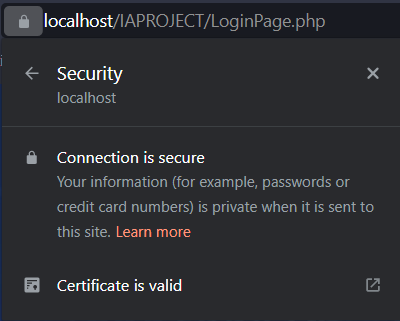


Fig 5. Secure connection

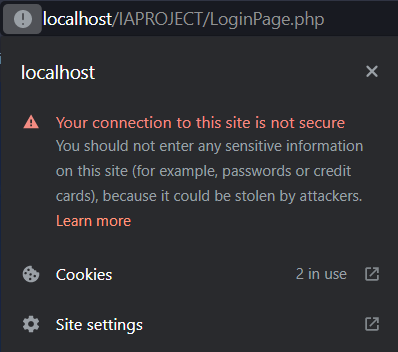


Fig 6. Unsecure connection

1. **Implementation**

For the implementation part of this project, we will divide the entire Project implementation in the following parts

* 1. **Back end**

The backend of this project relies on a database named encryption\_demo with two tables created in mysql.

The two tables are:

* Login
  + Stores the login details of users.
* Hashtable
  + Stores the additional details entered by the user.
  + Stores ciphertext(encrypted data) of the salary entered by user.

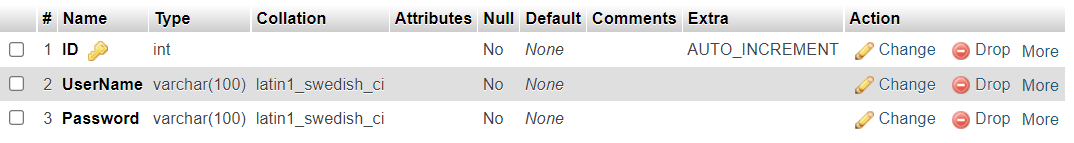


Fig 7. Structure of login table

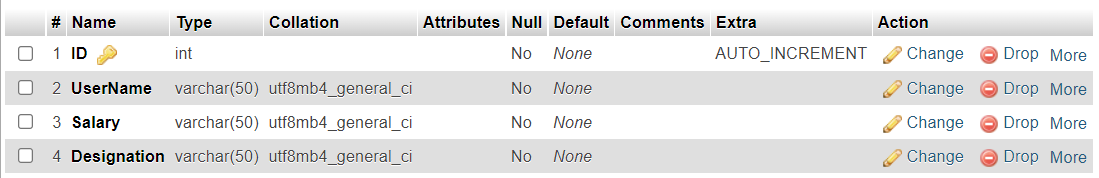


Fig 8. Structure of hashtable

* 1. **Front end**

For the front end of this project we have two webpages

* LoginPage
  + Allows user to login and access the homepage
* HomePage
  + Allows user to enter additional details like salary and designation



Fig 9. LoginPage

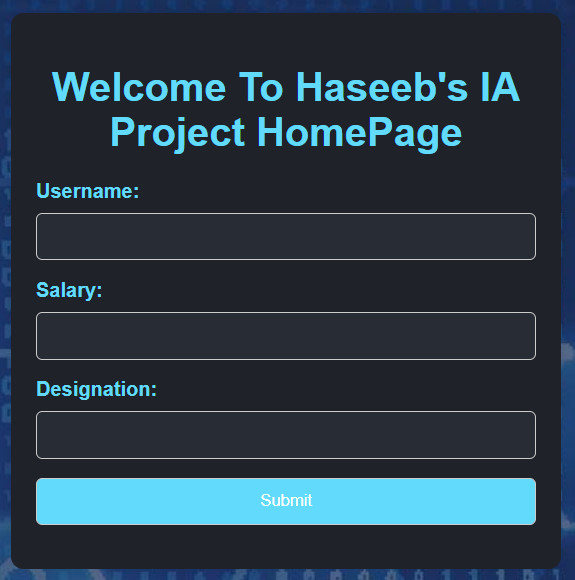


Fig 10. HomePage

* 1. **Code**

The code screenshots for both webpages are attached below

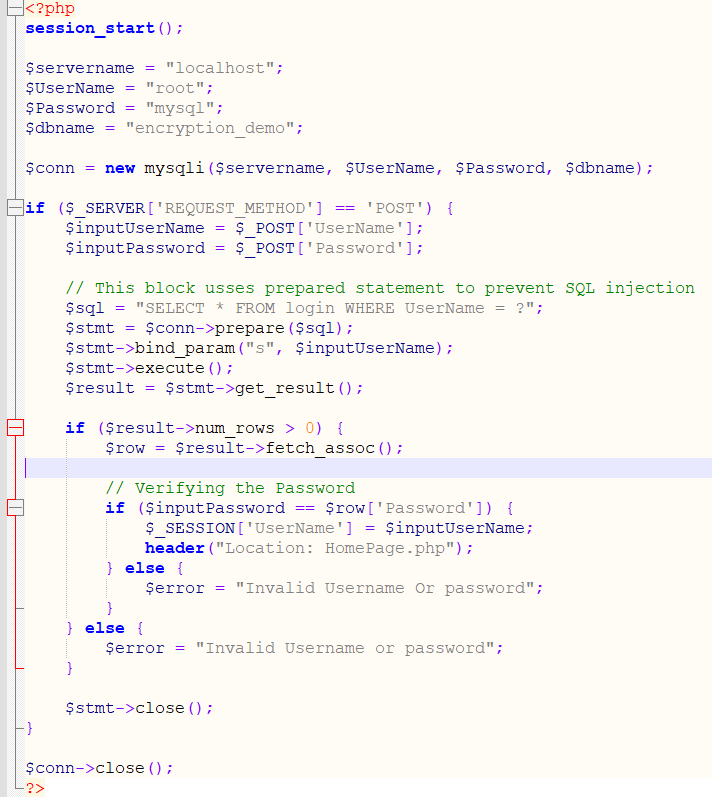
 

Fig 11. LoginPage Fig 12. HomePage code

* 1. **Security**
     1. **HTTPs**

Https is a secure protocol that runs on port 443. The use of Https allows us to securely communicate between front end and back end.

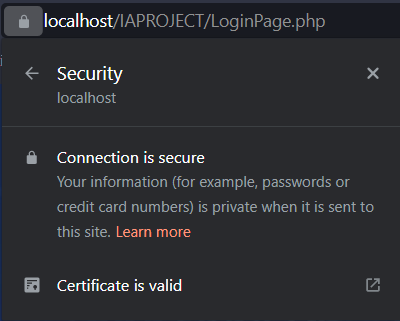


Fig 13. Implementation of HTTPs

* + 1. **Encryption**

In this project we used Advanced encryption standard (AES-256) to encrypt the salary entered by a user. AES-256 is an encryption technique that generates a random 256 bit hash value for plaintext.

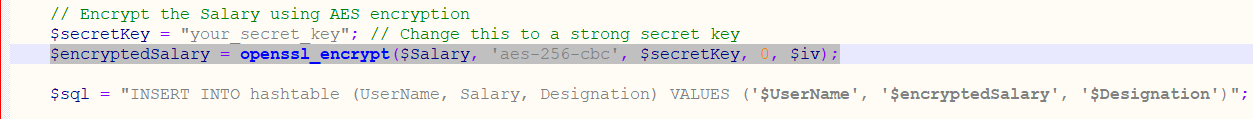


Fig 14. Encryption using AES-256

* + 1. **SQL injection countermeasures**

In this project we have also implemented countermeasures against SQL injection attacks by using a prepared statement and at the same time also validating the password field.

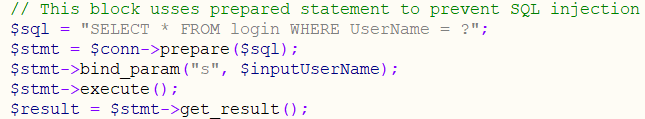


Fig 15. SQL injection countermeasures

* 1. **Testing**

Testing of the project includes trying login using invalid credentials.

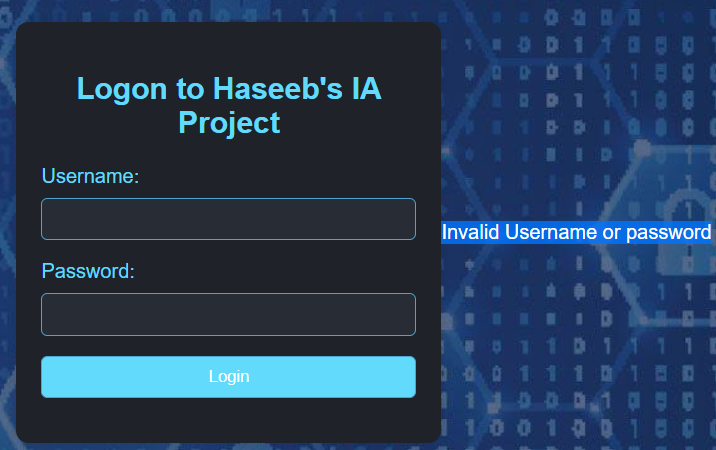


Fig 16. Error on Login Page

1. **Conclusion**

To sum up, the safe Web-Database Programming Project has focused on creating a framework for safe connection between a client and server, specifically utilizing SSL to ensure secure data exchange. The procedure comprised setting up a secure database, configuring the web server, and adding SSL certificates to guarantee communication secrecy. The computer code demonstrated a straightforward login process that integrated with the MySQL database and a home page that protected sensitive user information using AES encryption.

1. **Future improvements:**

Prospective Improvements: As time goes on, there will be chances to improve the project's functionality and security. Role-based access control (RBAC) and multi-factor authentication (MFA) can be used to improve user login security and provide more accurate control over user permissions. A framework from NIST can also be used to improve the security. Proactively strengthening the defence against changing threats will need regular security audits, in-depth code reviews, and keeping up with new security standards. Thoroughly examining UI improvements, investigating cutting-edge encryption methods, and using new technology will all help create a more durable and reliable online service.