**A-Pi Locker**

**Project Description:**

   A-Pi Locker is a secure record/password manager with the help of raspberry pi zero which act as a server device to store and access the encrypted information from the client device. I am using  RESTful API to transfer the encrypted information from the server to client and vice versa. It relies on a stateless, client-server, cacheable communication protocol—and in virtually all cases, the HTTP protocol is used. In server side I will use ReSTful API which is written in Google’s Go language which reads  and writes the encrypted record/password on SQLite databases. On the client side I run on HTML5 and JavaScript and it encrypts and decrypts the password/records using the CryptoJS for AES encryption and PRNG.

Also I will be modifying the configuration of Raspberry Pi Zero to enable ethernet connectivity over its USB port . It will create a new network interface when connected to the client (like other computers or smartphones) and we can connect with the server through any browser that supports HTML5.

**Tool Uniqueness**

 The Uniqueness of our tool is that it encrypts the data with AES encryption algorithm and it could be accessed from anywhere. It could be used with every possible devices that supports HTML5 in browser. I do not require any cloud. We need not pay for any server. It is cheaper and portable and can also be connected to smartphones.

**Programming Language:**

I am using Google’s Go language- it is a statically typed , compiled language in the tradition of C, with memory safety, garbage collection, structural typing and CSP-style concurrency. [there was a question to mention about the libraries that I would be using in our project, I have not answered it as GO language is really very new for me and yes I would be updating about it very shorty as I are yet in the process of learning.]

The hardware which I will be using is Raspberry Pi Zero  as I felt it more convenient and also cheaper.

**Test Environment:**

The test environment using here is Unity🡪 it is a graphical shell for the GNOME desktop environment .

**Story Board:**

**Server**  **Client**

**SQLite**

**CryptoJS**

**ReSTful API**

**PRNG**

**Raspberry Pi Zero** **other computer/smartphones**

**CLIENT**

**CSS**

**HTML5**

**CryptoJS \*\*\* PRNG**

**SERVER**

**GO Lang**