Using Client-Side Substitution Ciphers to Enhance Password Strength

Technical Manual

Student: Kevin Davitt
Supervisor: James Egan
C00215181@itcarlow.ie

Abstract

The purpose of this project is to develop a fully-fledged Web Browser extension that enables users to use previously insecure password practices that provide convenience, without the security concerns. This is achieved using a substitution cipher that operates client-side after the user has inputted the short/recycled/reused/common password that they remember. This first password can be though of as a seed word for the substitution cipher. Each service a user wants to log into will have a separate cipher assigned to it, meaning that the user may have to only remember a single password, when in actual fact, that single password is changed to numerous different ones, different for each service.

Contents

Introduction

The below contains all the code necessary to get the application to run, XAMPP or similar software should be used to locally host the project. However, there is currently a version running and mappa.ie that can be used. Some initial configuration is required to set up the databases and the AJAX and relative paths would need to be altered to reflect the chosen directory structure. The project hosted at mappa.ie has been configured with TLS, Digital Certificates, Load Balancing and the Web Server has been hardened. The required configuration for these tasks have been omitted from the technical manual as it is not necessary to get the project functioning locally.

Some style code has been omitted such as the code pulled in from bootstrap and FontAwesome (for icons), also omitted are the images and custom designed logos and icons for the user interface. However all of these are contained within the final project code zip file.

The Code is broken down into sections. Firstly, Web Server code is broken down into PHP, JavaScript and Style code. The PHP code is broken down per php file used along with its purpose on the Web server, similar explanations are taken for the other types of file. Web Extension code is all contained within a single section, with explanation as to the purpose of each file. The inner workings of functions are not explained algorithmically in this manual but all important functions are listed here and should be moderately easy to follow after reading. For example, the custom substitution cipher generation algorithm within genMap.php

Database Creation Code

DatabaseCreate.sql

```sql
CREATE USER 'mappauser'@'localhost' IDENTIFIED BY '<PASSWORD>';
CREATE DATABASE mappa;
Grant all privileges on mappa.* to 'mappauser'@'localhost';
/*--create Map Table*/
USE mappa;
CREATE TABLE map ( id smallint unsigned not null auto_increment, map varchar(455),
domain varchar(128), username varchar(64) not null, constraint pk_example primary key (id)) ;

/*--Create Session Table*/
CREATE TABLE sessions ( sessionid varchar(64), username varchar(64), csrf_token varchar(64),
successful_login_time varchar(20), last_click_time varchar(20), constraint pk_example primary key (sessionid)) ;

/*--Create BruteForceProtectionTable*/

CREATE TABLE bruteforceprotection ( hashed_ip_user_agent varchar(64), attempts smallint,
allowed_login_time varchar(20), allowed_signup_time varchar(20), last_active_time varchar(20),
device_csrf_token varchar(64) );
/*--Create Accounts Table; */
CREATE TABLE accounts ( username varchar(64), email varchar(96), account_password varchar(64),
salt varchar(128), creation_date varchar(20), max_maps int, constraint pk_example primary key (username)) ;

/*--Create logging Table; */
CREATE TABLE logging ( event_time varchar(64), event_description varchar(128),
outcome varchar(32), Hashed_IP_User_Agent varchar(64)) ;
```

This code is used to create the required tables in the database, the password has been removed.

Web Server PHP Code

WebsiteSecurityFunctions.php

This code contains all security and database related functions required for operation of the website and web extension calls.

```
<?php
 require_once 'header.php';


###### ###debugging
ini_set('display_errors',1);
ini_set('display_startup_errors',1);
error_reporting(E_ALL);


 ##########end debugging, delete this
##Function to SanitizeUserInput
function Sanitize($string)
{

      $badInput = array("<",      ">",  "&",     "",     """,     "/",  "{",   "}",   ";",  "\\", "(", ")", "$");
      $badInputAlts = array("&lt;","&gt;","&amp;", "&quot;", "&#x27;", "&#x2F;","&#x7B;","&#x7D;", "&#x3B;", "&#x5C;","&#x28;", "&#x29;", "&#x24;");
      #$string = "I am a <>{} test";

      $returnString = "";
      $length = strlen($string);
      for ($i=0; $i<$length; $i++) ##get each character in input
         {
            #echo "<BR>";
            $currentChar = $string[$i];
            $badCharFlag = False;
            $escapePos = -1;
            for ($j=0; $j<count($badInput); $j++)  ##check against each bad character
            {
               $badChar = $badInput[$j];
               if($currentChar == $badChar) #if char is bad
               {
                  $escapePos = $j;
                  #Echo "Bad Char Found at Position $i";
                  $badCharFlag = True;
               }
            }

            if($escapePos >= 0 && $escapePos < count($badInputAlts))
```

```
            {
                $currentChar = $badInputAlts[$escapePos];
                $badCharFlag = false;
            }

            if($badCharFlag == False) #if not a bad char, (IS GOOD CHAR) add to string
            {
                #echo "Good Char '$currentChar'";
                $returnString = "$returnString$currentChar";
            }
        }
      #echo "<br><br> OUTPUT $returnString";
   #$this->sanitisedText = $returnString;
   return $returnString ;

}

#Function to Hash Password with Salt
function PasswordHash($stringToHash, $salt)
{

   $iterations = 1000;
   $hash = hash_pbkdf2("sha256", $stringToHash, $salt, $iterations);
   return $hash;

}
#Function to create Session IDs
function generateRandomSessionID()
      {
         $sessionID = random_bytes(32);
         $sessionID = (bin2hex($sessionID));
         $sessionID = hash("sha256", $sessionID);
         return $sessionID;
      }

#Function to GenerateSalt
function generateRandomSalt()
      {
         $salt = random_bytes(32);
         $salt = (bin2hex($salt));
         return $salt;
      }

#Function to get the time
function getCurrentTimeString()
```

```
{
   date_default_timezone_set('Europe/Dublin');
$date = date('m/d/Y H:i:s ', time()); #in form 11/26/2019 02:58:10
return $date;
}
#Function to get the time from now offset in seconds
function getCurrentTimeStringOffset($offsetInSeconds)
{
   date_default_timezone_set('Europe/Dublin');
$date = date('m/d/Y H:i:s ', (time()+$offsetInSeconds)); #in form 11/26/2019 02:58:10
return $date;
}

##connect to DB
function connectToDB($start)
{
   $servername = "localhost";
   $username = "mappauser";
   $password = "r3NTdm00S!!!kcAL";
   $databaseName = "mappa";
      $conn = new mysqli($servername, $username, $password, $databaseName);// Create connection
   if($start == true)
   {
         if ($conn->connect_error)
         {
            die("ERROR: ");#               die("Connection failed: " . $conn->connect_error);
         }
      else
         {
            $databaseAvailable = true;
            return $conn;
         }
   }
   else
   {
      mysqli_close($conn);
   }
}

##check user exists
##check username exists
function checkUsernameExists($username)
{
   $retrievedUsername = "0";
   $conn = connectToDB(true); #connect db
```

```
    if ($conn->connect_error) #ensure it worked
            {
                die("Connection failed: " . $conn->connect_error);
            }
    else #successfully connected to database
    {

        $stmt = $conn->prepare("SELECT username FROM accounts WHERE username = ?");
        $stmt->bind_param("s", $username);
        $stmt->execute();
        $result = $stmt->get_result();
        if($result->num_rows === 0)
            {
                return $retrievedUsername ;
                exit(''); #<br>Incorrect username or password - no rows for name and password'
            }
        while($row = $result->fetch_assoc())
            {
                $retrievedUsername = $row['username'];
            }
        #echo "<br>RESULT: $retrievedPassword <br>" ;


    }
    connectToDB(false); #disconnect db
    if ($retrievedUsername == "0")
    {
        return false;
    }
    else
    {
        return true ;
    }
}
    ##insert new user, MUST PASS GOOD DATA
    function createNewUser($username, $emailaddress, $password)
    {
    $success = false ;
    $defaultMaxMaps = 50;
    $salt = generateRandomSalt();
    $time = getCurrentTimeString();
    $hashedPassword = PasswordHash($password,$salt) ;
    $conn = connectToDB(true); #connect db
    if ($conn->connect_error) #ensure it worked
            {
```

```
            die("Connection failed: " . $conn->connect_error);
        }


    else #successfully connected to database
    {
        $emptySession = "-";
        $stmt = $conn-
>prepare("INSERT INTO accounts ( username, email, account_password, salt, creation_date, max_maps ) VA
LUES (  ?, ?, ?, ?, ?, ? );");
        $stmt-
>bind_param("sssssi", $username, $emailaddress, $hashedPassword, $salt, $time, $defaultMaxMaps);
        $success = $stmt->execute();

        connectToDB(false); #disconnect db

    }


    return $success ; #returns whether insert was sucessful or not.

    }


    ###getSaltFromUsername
    function getSaltFromUsername($username)
{
    $retrievedSalt = "0";
    $conn = connectToDB(true); #connect db
    if ($conn->connect_error) #ensure it worked
        {
            die("Connection failed: " . $conn->connect_error);
        }
    else #successfully connected to database
    {

        $stmt = $conn->prepare("SELECT salt FROM accounts WHERE username = ?");
        $stmt->bind_param("s", $username);
        $stmt->execute();
        $result = $stmt->get_result();
        if($result->num_rows === 0)
        {
            return $retrievedSalt ;
            exit(''); #<br>Incorrect username or password - no rows for name and password'
        }
```

```
        while($row = $result->fetch_assoc())
            {
                $retrievedSalt = $row['salt'];
            }
        #echo "<br>RESULT: $retrievedPassword <br>" ;
        connectToDB(false); #disconnect db

    }
    if($retrievedSalt == "0")
    {
        return false;
    }
    else #found the salt, return the salt
    {
        return $retrievedSalt ;
    }

}

###validateCredentials
function validateCredentials($username,$password)
{
    $retrievedPassword = "0";
    $conn = connectToDB(true); #connect db
    if ($conn->connect_error) #ensure it worked
            {
                die("Connection failed: " . $conn->connect_error);
            }
    else #successfully connected to database
    {

        $stmt = $conn->prepare("SELECT account_password FROM accounts WHERE username = ?");
        $stmt->bind_param("s", $username);
        $stmt->execute();
        $result = $stmt->get_result();
        if($result->num_rows === 0)
            {
                return $retrievedPassword ;
                exit(''); #<br>Incorrect username or password - no rows for name and password'
            }
        while($row = $result->fetch_assoc())
            {
                $retrievedPassword = $row['account_password'];
            }
        #echo "<br>RESULT: $retrievedPassword <br>" ;
```

```
        connectToDB(false); #disconnect db

    }
    if($retrievedPassword == $password) #passwords match, valid login
    {
        return true;
    }
    else #passwords do not match, invalid login attempt
    {
        return false ;
    }

}

##
##### CREATE SESSION
##update session cookie in database
function createSession($username) #EXPECTS VALID INPUT!!
{
    //enter session cookie into database
            $conn = connectToDB(true); #connect db


                #generateSessionID
                $session = generateRandomSessionID();
                while(CheckAuthenticatedSession($session) !== false) #ensure no dupes
                {
                    $session = generateRandomSessionID();
                }
                $csrf = generateRandomSessionID();
                $time = getCurrentTimeString();
                #create sessionID in DB and associate it with the user logging in
                $stmt = $conn-
>prepare("INSERT INTO sessions (sessionid, username, csrf_token, successful_login_time, last_click_time)
VALUES (?, ?, ?, ?, ?);");
                $stmt->bind_param("sssss", $session, $username, $csrf, $time, $time);
                $stmt->execute();

                ###DO BRUTE FORCE PROTECTION
                $deviceHash = getDeviceHash() ;
                $zero = 0 ;
                $stmt = $conn-
>prepare("update bruteforceprotection set attempts = ? WHERE hashed_ip_user_agent = ?");
                $stmt->bind_param("is",$zero, $deviceHash);
                $stmt->execute();
```

```
        connectToDB(false); #disconnect db


        #set cookie in users browser
        $cookie_name = "AUTHENTICATED";
        #$localIP = getHostByName(getHostName());
        #ECHO "<BR> LOCALIP: $localIP";
        setcookie($cookie_name, $session, time() + 3600, '/',NULL,NULL,1 ); #3600 for one hour,

#####Csyntax setcookie($name, $value, $expire, $path, $domain, $secure, $httponly )
}

####DESTROY SESSION
function destroySession($sessionID)
{
   //enter session cookie into database
          $conn = connectToDB(true); #connect db



             $session = "-";

             #update sessionID in DB, set to "-" (not logged in)
             $stmt = $conn->prepare("delete FROM sessions WHERE sessionid = ?");
             $stmt->bind_param("s", $sessionID);
             $stmt->execute();

             ##set attempts for device to 0, brute force


        connectToDB(false); #disconnect db


        #set cookie in users browser to delete (set to past expiry)
        $cookie_name = "AUTHENTICATED";
        setcookie($cookie_name, $session, time() - 3600, "/",null,null,1); #-3600 for one hour expire 1 hour ago
}

##ASSESS LOGIN STATUS
function CheckAuthenticatedSession($sessionID)
{
   $retrievedUser = "0";
   $conn = connectToDB(true); #connect db
   if ($conn->connect_error) #ensure it worked
          {
```

```
            die("Connection failed: " . $conn->connect_error);
        }
    else #successfully connected to database
    {

        $stmt = $conn->prepare("SELECT username FROM sessions WHERE sessionid = ?");
        $stmt->bind_param("s", $sessionID);
        $stmt->execute();
        $result = $stmt->get_result();
        if($result->num_rows === 0)
            {
                return false;
                #exit(''); #<br>Incorrect username or password - no rows for name and password'
            }
        while($row = $result->fetch_assoc())
            {
                $retrievedUser = $row['username'];
            }
        #echo "<br>RESULT: $retrievedPassword <br>" ;
        connectToDB(false); #disconnect db

    }

    if($retrievedUser != "0") #passwords match, valid login
    {
        return $retrievedUser;
    }
    else #invalid session
    {
        return false ;
    }
}
####
function validateCookieStructure($sessionID)
{
    $badAuthCookie = false;
    $cookieLen = strlen($sessionID);
    $i = 0;
    $validValues = "1234567890abcdef";
    while($i < $cookieLen && $badAuthCookie == false)
    {
        $currentChar = $sessionID[$i];
        if(strpos( $validValues, $currentChar ) !== false)
        {
```

```
        }
        else #sessionID contains invalid charss.
            {
                #echo "INVALID CHAR: $currentChar";
                $badAuthCookie = true ;
            }
        $i++ ;
    }
    return $badAuthCookie ;
}



###GET USERNAME FROM SESSION ID
##get session ID
function getUsernamefromSessionID($SessionID)
{
    $retrievedUsername = "0";
    if($SessionID == "-") #Default session val in database, if this is passed, find no sessions
    {
    }
    else
    {
        $conn = connectToDB(true); #connect db
        if ($conn->connect_error) #ensure it worked
                {
                    die("Connection failed: " . $conn->connect_error);
                }
        else #successfully connected to database
        {

            $stmt = $conn->prepare("SELECT username FROM sessions WHERE sessionid = ?");
            $stmt->bind_param("s", $SessionID);
            $stmt->execute();
            $result = $stmt->get_result();
            if($result->num_rows === 0)
                {
                    return $retrievedUsername ;
                    exit(''); #<br>Incorrect username or password - no rows for name and password'
                }
            while($row = $result->fetch_assoc())
                {
                    $retrievedUsername = $row['username'];
                }
            #echo "<br>RESULT: $retrievedPassword <br>" ;
```

```
            connectToDB(false); #disconnect db

        }
    }

    return $retrievedUsername ;
}

####ADD NEW MAP FUNCTION

function AddNewMap($username, $mapName, $mapString)
    {
        {
            $success = false ;
            $conn = connectToDB(true); #connect db
            if ($conn->connect_error) #ensure it worked
                    {
                        die("Connection failed: " . $conn->connect_error);
                    }


            else #successfully connected to database
            {
                $emptySession = "-";
                $stmt = $conn-
>prepare("INSERT INTO map (id, map, domain, username) VALUES (null, ?,?,?) ;");
                $stmt->bind_param("sss", $mapString, $mapName, $username);
                $success = $stmt->execute();

                connectToDB(false); #disconnect db

            }

            unset($secFeatures); #destroy sec object
            return $success ; #returns whether insert was sucessful or not.

            }
    }


#END ADD NEW MAP FUNCTION

##CHECK MAP NAME ALREADY EXISTS
function checkMapNameExists($mapName,$username)
{
```

```
    $retrievedMapName = "0";
    $conn = connectToDB(true); #connect db
    if ($conn->connect_error) #ensure it worked
            {
                die("Connection failed: " . $conn->connect_error);
            }
    else #successfully connected to database
    {

        $stmt = $conn->prepare("SELECT domain FROM map WHERE domain = ? and username = ?");
        $stmt->bind_param("ss", $mapName, $username);
        $stmt->execute();
        $result = $stmt->get_result();
        if($result->num_rows === 0)
            {
                return $retrievedMapName ;
                exit(''); #<br>Incorrect username or password - no rows for name and password'
            }
        while($row = $result->fetch_assoc())
            {
                $retrievedMapName = $row['domain'];
            }
        #echo "<br>RESULT: $retrievedPassword <br>" ;


    }
    connectToDB(false); #disconnect db
    if ($retrievedMapName == "0")
    {
        return false;
    }
    else
    {
        return true ;
    }
}
##END CHECK MAP NAME ALREADY EXISTS

###GET NUMBER OF USERS MAPS
function GetNumberOfUsersMaps($username)
        {

            $count = 0;
                $conn = connectToDB(true); #connect db
```

```
            #update sessionID in DB, set to "-" (not logged in)
            $stmt = $conn->prepare("select map from map WHERE username = ?");
            $stmt->bind_param("s",$username);
            $stmt->execute();
            $result = $stmt->get_result();
          while($row = $result->fetch_assoc())
          {
            $count++ ;
          }

        connectToDB(false); #disconnect db
        return $count;
      }
###END GET NUMBER OF USERS MAPS
##GET USERS MAX ALLOWED MAPS
function GetMaxUserMaps($username)
      {

        $retrievedMaxMaps = "0";
          $conn = connectToDB(true); #connect db




            #update sessionID in DB, set to "-" (not logged in)
            $stmt = $conn->prepare("select max_maps from accounts WHERE username = ?");
            $stmt->bind_param("s",$username);
            $stmt->execute();
            $result = $stmt->get_result();
          while($row = $result->fetch_assoc())
          {
            $retrievedMaxMaps = $row['max_maps'];
          }

        connectToDB(false); #disconnect db
        return $retrievedMaxMaps;
      }
##END GET USERS MAX ALLOWED MAPS
#time comparison
function getDifferenceBetweenDates($closeTime, $farTime)
  {
    $dateOneSeconds = strtotime($farTime);
```

```
    $dateTwoSeconds = strtotime($closeTime);

    $returnVal = ($dateTwoSeconds -$dateOneSeconds);
    #echo "<br>Difference in seconds: $diff" ;
    return $returnVal ;
  }
##end time comparison
###CheckSessionIssueTimeValid
function CheckSessionIssueTimeValid($sessionid) #GETS SESSION ISSUE TIME AND DESTROYS SESSI
ON IF INVALID
    {

      $retrievedSessionIssueTime = false;
        $conn = connectToDB(true); #connect db




          $stmt = $conn->prepare("select successful_login_time from sessions WHERE sessionid = ?");
          $stmt->bind_param("s",$sessionid);
          $stmt->execute();
          $result = $stmt->get_result();
        while($row = $result->fetch_assoc())
        {
          $retrievedSessionIssueTime = $row['successful_login_time'];
        }

        $nowTime = getCurrentTimeString();
        $diff = getDifferenceBetweenDates($nowTime, $retrievedSessionIssueTime); #if positive value, des
troy session
        #echo "TIME DIFF: $diff" ;
        $validLoginDuration = 3600; #SESSIONS LAST 1 HOUR
        if($diff > $validLoginDuration && $retrievedSessionIssueTime != false )
        {
          echo "DESTROYING SESSION";
          $stmt = $conn->prepare("delete from sessions WHERE sessionid = ?");
          $stmt->bind_param("s",$sessionid);
          $stmt->execute();
          $result = $stmt->get_result();

          setcookie($sessionID_Cookie_Name, NULL, time() - 3600, "/",null,null,1); #-
3600 for one hour expire 1 hour ago
          #reload page
          header("Refresh:0");
          die();
```

```
            }

        connectToDB(false); #disconnect db
        return $retrievedSessionIssueTime;
    }

###END CheckSessionIssueTimeValid
###GET REMAINING SESSION TIME FOR COOKIE
###get absolutetimeout remaining seconds

##END GET REMIANING SESSION TIME FOR COOKIE
function getRemainingSessionTime($validSessionID)
{

    $retrievedTimeoutTime = "0";
        $conn = connectToDB(true); #connect db

        $time =getCurrentTimeString();



            #update sessionID in DB, set to "-" (not logged in)
            $stmt = $conn->prepare("select successful_login_time from sessions WHERE sessionid = ?");
            $stmt->bind_param("s", $validSessionID);
            $stmt->execute();
            $result = $stmt->get_result();
        while($row = $result->fetch_assoc())
        {
            $retrievedTimeoutTime = $row['successful_login_time'];
         }
          #ECHO "<BR>(Get remaining Session time function) <br>currenttime: $time <br>retrievedTime: $ret
rievedTimeoutTime";
    $difference = getDifferenceBetweenDates($time, $retrievedTimeoutTime);


    connectToDB(false); #disconnect db



    return $difference;
}
###UPDATE SESSION ID
function updateSessionID($validSessionID)
  {
    $conn = connectToDB(true); #connect db
```

```
            #generateSessionID
            $newSessionID = generateRandomSessionID();
            while(CheckAuthenticatedSession($newSessionID) !== false) #ensure no dupes
            {
               $newSessionID = generateRandomSessionID();
            }
            $time = getCurrentTimeString();
            #update sessionID in DB
            $stmt = $conn->prepare("update sessions set sessionid = ? WHERE sessionid = ?");
            $stmt->bind_param("ss", $newSessionID, $validSessionID);
            $stmt->execute();
            $remainingSeconds = getRemainingSessionTime($newSessionID);
            #echo "(function updateSessionID) Remaining Seconds: $remainingSeconds";
            ##update cookie
            $cookie_name = "AUTHENTICATED";
            setcookie($cookie_name, $newSessionID, (time() + (3600-
$remainingSeconds)), "/", null, null, 1); #add the remaining time in current session, 1hr - elapsed time in sessio
n so far
            connectToDB(false); #disconnect db
            return $newSessionID;

   }
###END UPDATE SESSION ID


###delete all data in user account
function destroyUserData($username)
{
   //enter session cookie into database
   $mapDelete = false;
   $accDelete = false;
   $sessionDelete = false;
         $conn = connectToDB(true); #connect db



            #delete from map table
            $stmt = $conn->prepare("delete FROM map WHERE username = ?");
            $stmt->bind_param("s", $username);
            $mapDelete = $stmt->execute();
```

```
            #delete from acc table
            $stmt = $conn->prepare("delete FROM accounts WHERE username = ?");
            $stmt->bind_param("s", $username);
            $accDelete = $stmt->execute();



            #delete from session table
            $stmt = $conn->prepare("delete FROM sessions WHERE username = ?");
            $stmt->bind_param("s", $username);
            $sessionDelete = $stmt->execute();







      connectToDB(false); #disconnect db


      #set cookie in users browser to delete (set to past expiry)
      $cookie_name = "AUTHENTICATED";
      setcookie($cookie_name, NULL, time() - 3600, "/",null,null,1); #-3600 for one hour expire 1 hour ago

      if($mapDelete == true && $accDelete == true && $sessionDelete == true  )
         {
            return true;
         }
      else
         {
               return false ;
         }
}


###end delete all data on user

##get device hash
##get device hash
function getDeviceHash()
   {
      $userAgent =  Sanitize( $_SERVER['HTTP_USER_AGENT']);
      $remoteIP =  Sanitize( $_SERVER['REMOTE_ADDR']);
      $together = "$userAgent$remoteIP";
      $hashOfDevice = hash("sha256", $together );
```

```
        return $hashOfDevice;
    }


##end get device hash

##checkallowed signup
function checkDeviceAllowedSignUp()
{
    $allowedSignup = false ;
    $deviceHash = getDeviceHash() ;
    $allowedSignupTime = "0";

    $nowTime = getCurrentTimeString();  #add 5 mins to current time
    $conn = connectToDB(true);
    $stmt = $conn-
>prepare("select allowed_signup_time from bruteforceprotection WHERE hashed_ip_user_agent = ?");
        $stmt->bind_param("s", $deviceHash);
        $stmt->execute();
        $result = $stmt->get_result();
        while($row = $result->fetch_assoc())
        {
            $allowedSignupTime = $row['allowed_signup_time'];
        }
    $timeDiff = getDifferenceBetweenDates($nowTime, $allowedSignupTime); ##param 1 - param 2
    $allowedSignup = $timeDiff;
    #echo "<BR>$allowedSignupTime, $nowTime, ALLOWED SIGNUP : $timeDiff";
    if($timeDiff >= 0) #if allowed signup time is in the past, you can sign up now
    {
        $allowedSignup = true ;
        #echo "TRUE" ;
    }
    else
    {
        $allowedSignup = $allowedSignup*-1; ##MAKE POSITIVE
        #echo "FALSE" ;
    }

    connectToDB(false) ; #disconnect db
    return $allowedSignup; #
}

##end check allowed sign up
###################################
##get device allowed sign in
```

```php
function checkDeviceAllowedSignIn()
{
    $allowedSignup = false ;
    $deviceHash = getDeviceHash() ;
    $allowedSignupTime = "0";

    $nowTime = getCurrentTimeString();  #add 5 mins to current time
    $conn = connectToDB(true);
    $stmt = $conn-
>prepare("select allowed_login_time from bruteforceprotection WHERE hashed_ip_user_agent = ?");
        $stmt->bind_param("s", $deviceHash);
        $stmt->execute();
        $result = $stmt->get_result();
        while($row = $result->fetch_assoc())
        {
            $allowedSignupTime = $row['allowed_login_time'];
        }
    $timeDiff = getDifferenceBetweenDates($nowTime, $allowedSignupTime); ##param 1 - param 2
    $allowedSignup = $timeDiff;
    #echo "<BR>$allowedSignupTime, $nowTime, ALLOWED LOGIN : $timeDiff";
    if($timeDiff >= 0) #if allowed signup time is in the past, you can sign up now
    {
        $allowedSignup = true ;
        #echo "TRUE" ;
    }
    else
    {
        $allowedSignup = $allowedSignup*-1; ##MAKE POSITIVE
        #echo "FALSE" ;
    }

    connectToDB(false) ; #disconnect db
    return $allowedSignup; #
}
##end get device allowed login

##create user agent track
function createUserAgentTrack()
{

    $retrievedDeviceHash = "0";
    $conn = connectToDB(true); #connect db
    $hashOfDevice = getDeviceHash() ;
        if ($conn->connect_error) #ensure it worked
            {
```

```
            die("Connection failed: " . $conn->connect_error);
        }
    else #successfully connected to database
        {

            $stmt = $conn-
>prepare("SELECT hashed_ip_user_agent FROM bruteforceprotection WHERE hashed_ip_user_agent = ?");
            $stmt->bind_param("s", $hashOfDevice);
            $stmt->execute();
            $result = $stmt->get_result();

            while($row = $result->fetch_assoc())
                {
                    $retrievedDeviceHash = $row['hashed_ip_user_agent'];
                }
        }
  $alreadyInDB = false;
  $lastActiveTime = getCurrentTimeStringOffset(0); #ensure time in past so users can login right away
  $time = getCurrentTimeStringOffset(-10); #ensure time in past so users can login right away
  $signup_time = getCurrentTimeStringOffset(-10); #ensure time in past so users can sign up right away
  $csrf = generateRandomSessionID() ;
  if(strlen($retrievedDeviceHash) != 64) ##entry for useragent not in db
    {
      #echo "<br> Entering Device into tracking DB" ;

    ##insert the current block
    $zero = 0;
    $stmt = $conn-
>prepare("INSERT INTO bruteforceprotection ( hashed_ip_user_agent, attempts, allowed_login_time,allowed
_signup_time, last_active_time, device_csrf_token ) VALUES (?, ?, ?,?,?,? );");
    $stmt->bind_param("sissss", $hashOfDevice,$zero,$time,$signup_time, $lastActiveTime, $csrf );
    $success = $stmt->execute();
    $alreadyInDB = true;

    }
  else
    {
      #echo "<br>  Device already exists in tracking DB" ;
      ##update last active time e.g code"update sessions set sessionid = ? WHERE sessionid = ?"
      #$stmt = $conn-
>prepare("UPDATE bruteforceprotection SET last_active_time = ? where hashed_ip_user_agent = ?;");
      #$stmt->bind_param("ss",  $lastActiveTime, $hashOfDevice );
      #$success = $stmt->execute();

    }
```

```
        $conn = connectToDB(false); #connect db
        return $alreadyInDB;
}
##update device CSRF
Function UpdateDeviceCSRF()
{
    $conn = connectToDB(true); #connect db
    $hashOfDevice = getDeviceHash() ;
    $csrf = generateRandomSessionID() ;
    if ($conn->connect_error) #ensure it worked
            {
                die("Connection failed: " . $conn->connect_error);
            }
    else #successfully connected to database
    {

        $stmt = $conn-
>prepare("UPDATE bruteforceprotection SET device_csrf_token = ? where hashed_ip_user_agent = ?;");
        $stmt->bind_param("ss", $csrf, $hashOfDevice);
        $stmt->execute();
        $result = $stmt->get_result();

    }
    $conn = connectToDB(false); #connect db
}

##end update device csrf

##end create user agent track

###GET DEVICE CSRF
function GetDeviceCSRF()
{
    $retrievedDeviceCSRF = "0";
    $conn = connectToDB(true); #connect db
    $hashOfDevice = getDeviceHash() ;
    if ($conn->connect_error) #ensure it worked
            {
                die("Connection failed: " . $conn->connect_error);
            }
    else #successfully connected to database
    {

        $stmt = $conn-
>prepare("SELECT device_csrf_token FROM bruteforceprotection WHERE hashed_ip_user_agent = ?");
```

```
        $stmt->bind_param("s", $hashOfDevice);
        $stmt->execute();
        $result = $stmt->get_result();

        while($row = $result->fetch_assoc())
            {
                $retrievedDeviceCSRF = $row['device_csrf_token'];
            }

    }
    $conn = connectToDB(false);
    return $retrievedDeviceCSRF;
}
##END GET DEVICE CSRF

##log usersign up
function blockUserSignUp()
    {
        $conn = connectToDB(true); #connect db

        $hashOfDevice = getDeviceHash() ;
                #generateSessionID
                $allowedSignupInSeconds = 300 ; ##5 mins between allowed sign up
                $time = getCurrentTimeStringOffset($allowedSignupInSeconds);
                #update sessionID in DB
                $stmt = $conn-
>prepare("update bruteforceprotection set allowed_signup_time = ? WHERE hashed_ip_user_agent = ?");
                $stmt->bind_param("ss", $time, $hashOfDevice );
                $stmtSuccess = $stmt->execute();

                connectToDB(false); #disconnect db
                return $stmtSuccess;

    }
#end log usersign up

#prevent user logging in for time
##log usersign up
function blockUserSignIn()
    {
        $conn = connectToDB(true); #connect db

        $hashOfDevice = getDeviceHash() ;
                #generateSessionID
                $allowedSignupInSeconds = 300 ; ##5 mins between allowed sign up
```

```
            $time = getCurrentTimeStringOffset($allowedSignupInSeconds);
            #update sessionID in DB
            $stmt = $conn-
>prepare("update bruteforceprotection set allowed_login_time = ? WHERE hashed_ip_user_agent = ?");
            $stmt->bind_param("ss", $time, $hashOfDevice );
            $stmtSuccess = $stmt->execute();

            connectToDB(false); #disconnect db
            return $stmtSuccess;

    }
##end prevent use loggin in for time

##increment invalid login
function incrementInvalidLogin()
    {
        $conn = connectToDB(true); #connect db
           $retrievedAttempts = 1000;
        $hashOfDevice = getDeviceHash() ;
            ##get current attempts
            $stmt = $conn-
>prepare("select attempts from bruteforceprotection WHERE hashed_ip_user_agent = ?");
            $stmt->bind_param("s", $hashOfDevice );
            $stmtSuccess = $stmt->execute();
            $result = $stmt->get_result();
            while($row = $result->fetch_assoc())
               {
                  $retrievedAttempts = $row['attempts'];
               }
            $retrievedAttempts = $retrievedAttempts + 1;
            #update allowed login time in DB
            $stmt = $conn-
>prepare("update bruteforceprotection set attempts = ? WHERE hashed_ip_user_agent = ?");
            $stmt->bind_param("is", $retrievedAttempts, $hashOfDevice );
            $stmtSuccess = $stmt->execute();

            connectToDB(false); #disconnect db
            return $retrievedAttempts;

    }
#end increment invalid logins

##get current csrf token
function getCurrentCSRF($sessionID)
    {
```

```
    $conn = connectToDB(true); #connect db
      $retrievedCSRF = false;
        ##get current csrf token
        $stmt = $conn->prepare("select csrf_token from sessions WHERE sessionid = ?");
        $stmt->bind_param("s", $sessionID );
        $stmtSuccess = $stmt->execute();
        $result = $stmt->get_result();
        while($row = $result->fetch_assoc())
          {
            $retrievedCSRF = $row['csrf_token'];
          }



        connectToDB(false); #disconnect db
        return $retrievedCSRF;

  }

##end get current csrf token

##update csrf token
function updateCSRF($sessionID)
  {
    $conn = connectToDB(true); #connect db
      $newCSRF = generateRandomSalt();;

        $stmt = $conn->prepare("update sessions set csrf_token = ? WHERE sessionid = ?");
        $stmt->bind_param("ss", $newCSRF,$sessionID );
        $stmtSuccess = $stmt->execute();



        connectToDB(false); #disconnect db
        return $stmtSuccess;

  }

##end update csrf token

##destroy all users sessions

function destroyAllSessions($username)
{
  //enter session cookie into database
```

```
        $conn = connectToDB(true); #connect db



            $session = "-";

            #update sessionID in DB, set to "-" (not logged in)
            $stmt = $conn->prepare("delete FROM sessions WHERE username = ?");
            $stmt->bind_param("s", $username);
            $stmt->execute();




    connectToDB(false); #disconnect db


    #set cookie in users browser to delete (set to past expiry)
    $cookie_name = "AUTHENTICATED";
    setcookie($cookie_name, $session, time() - 3600, "/",null,null,1); #-3600 for one hour expire 1 hour ago
}

##end destroy all users sessions

##change password
function changePassword($username, $password)
  {
    $conn = connectToDB(true); #connect db

            $stmt = $conn->prepare("update accounts set account_password = ? WHERE username = ?");
            $stmt->bind_param("ss", $password,$username );
            $stmtSuccess = $stmt->execute();



            connectToDB(false); #disconnect db
            return $stmtSuccess;

  }
##end change password

##change email
function changeEmail($email, $username)
  {
    $conn = connectToDB(true); #connect db
```

```
            $stmt = $conn->prepare("update accounts set email = ? WHERE username = ?");
            $stmt->bind_param("ss", $email, $username );
            $stmtSuccess = $stmt->execute();



            connectToDB(false); #disconnect db
            return $stmtSuccess;

    }
##end change email




###INSERT LOG EVENT

 function logEvent($event, $actiontaken)
 {

    $success = false;
 $time = getCurrentTimeString();
 $deviceHash =getDeviceHash();

 $conn = connectToDB(true); #connect db
 if ($conn->connect_error) #ensure it worked
        {
            die("Connection failed: " . $conn->connect_error);
        }


 else #successfully connected to database
 {
    $emptySession = "-";
    $stmt = $conn-
>prepare("INSERT INTO logging ( event_time, event_description, outcome, Hashed_IP_User_Agent ) VALU
ES (  ?, ?, ?, ? );");
    $stmt->bind_param("ssss", $time, $event, $actiontaken, $deviceHash );
    $success = $stmt->execute();

    connectToDB(false); #disconnect db

 }
```

```
 return $success ; #returns whether insert was sucessful or not.

 }







###end insert log details
###delete map
function deleteMap($username, $mapname)
{
    //enter session cookie into database
        $conn = connectToDB(true); #connect db



            $success = false;

            #update sessionID in DB, set to "-" (not logged in)
            $stmt = $conn->prepare("delete FROM map WHERE domain = ? AND username = ?");
            $stmt->bind_param("ss", $mapname, $username);
            $success = $stmt->execute();

            ##set attempts for device to 0, brute force


      connectToDB(false); #disconnect db


      return $success;
}

##end delete map

##########################################################################################
########################################
#######################################################                ###################
#########################
####################################################   THIS CODE RUNS ON EVERY PAGE   #
##########################################
#######################################################                ###################
#########################
```

```
#########################################################################################################
#######################################
$authCookieSet = false ;
$badAuthCookie = false ;
$passedSessionID = "-";
$sessionID_Cookie_Name = "AUTHENTICATED";
$loggedInUsername = "-";
if(isset($_COOKIE['AUTHENTICATED'])) ##ensure old session id not being passed
      {
         $badAuthCookie = validateCookieStructure($_COOKIE[$sessionID_Cookie_Name]);
         if($badAuthCookie == true)
         {
            #block user here
            blockUserSignIn() ;
            blockUserSignUp();
            logEvent("Invalid chars in SessionID", "BLOCKED FOR 300 SECONDS");
            die("FAILED COOKIE STRUCTURE - BLOCKING USER") ;

         }
         $validSessionCheck = CheckSessionIssueTimeValid( $_COOKIE[$sessionID_Cookie_Name]);
         if($validSessionCheck == false)
         {
            #block user here
            blockUserSignIn() ;
            blockUserSignUp();
            echo "PASSED = " . $_COOKIE[$sessionID_Cookie_Name];
            logEvent("Invalid SessionID", "BLOCKED FOR 300 SECONDS");
            die("FAILED SESSION CHECK - BLOCKING USER") ;
         }
         $passedSessionID = $_COOKIE[$sessionID_Cookie_Name];
         $passedSessionID = updateSessionID($passedSessionID) ;##update the sessionid
      }

$validSession = false;
if(isset($_COOKIE[$sessionID_Cookie_Name])) #seesssion
   {

      $authCookieSet = true ;
      $validSession = true;
      $loggedInUsername= getUsernamefromSessionID($passedSessionID);
   }



   ####Track BruteForcing of sign up
```

```
    $addedToTable = createUserAgentTrack() ; #returns true if new device
    $allowedSignup = checkDeviceAllowedSignUp();
    if($allowedSignup === true)
    {
       #echo "DEVICE IS CURRENTLY ALLOWED SIGNUP" ;
    }
    else
    {
       $whenCanSignUp = "WEBSITE-SEC-FUNCS---
DEVICE CAN SIGN UP IN $allowedSignup seconds<BR>" ;
    }
    ####Track BruteForcing of sign in
    $allowedLogin = checkDeviceAllowedSignIn();

    if($allowedLogin === true)
       {
          #echo "DEVICE IS CURRENTLY ALLOWED LOG IN" ;
       }
    else
       {
          $whenCanLogIn ="WEBSITE-SEC-FUNCS---
DEVICE CAN LOG IN AFTER $allowedLogin seconds<BR>";
       }


       $retrievedCSRF = getCurrentCSRF($passedSessionID);


?>
```

Header.php

This is enforcing headers that are required for hardening of the Web server, these headers are already configured on the Apache server itself but is being reinforced here for redundancy.

```
<?php
header('X-Frame-Options: deny');
header('Content-Security-Policy: upgrade-insecure-requests');
header('X-XSS-Protection:1; mode=block');
header("Cache-Control: no-store, no-cache, must-revalidate, max-age=0");
header("Pragma: no-store,no-cache, must-revalidate");
header('X-Content-Type-Options: nosniff');
header_remove("X-Powered-By");
?>
```

Home.php

This is the home page that the user is greeted with on visiting the website, you can see that cross origin resources such as bootstrap are being pulled in. The integrity of these files is first being checked, to ensure that in the event of supply chain compromise the scripts will not be used.

```
<html>

<?php
$num = rand()  ;
?>
<head>
<meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="description" content="Description of mappa here"></head>
  <title>Mappa</title>
 <link href="./fontAwesome/css/all.css" rel="stylesheet">

<link rel="stylesheet" href="./fontAwesome/css/all.css" />

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">
<script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="anonymous"></script>
<script src="./Scripts/home_typewriter_scripts.js" > </script>
<script src="https://kit.fontawesome.com/eca0e10f93.js" crossorigin="anonymous"></script>

<?php
echo "<link rel='stylesheet' href='./Styles/style.css?id=$num'";
?>
<STYLE>


</STYLE>
</head>
<body>

<!-- NAV BAR !-->
```

```
<nav>
<section class="nav-bar">
<nav class="navbar navbar-expand-lg navbar-light">
 <a class="navbar-brand" href="#">
   <?php echo
   "<img src='./imgs/mappa_logo.svg?id=$num>'";
   ?>
  </a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-
controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
   <i class="fas fa-bars" style="color:#DDDDDD;"></i>
  </button>
  <div class="collapse navbar-collapse" id="navbarNav">
   <ul class="navbar-nav ml-auto">
     <li class="nav-item">
       <a class="nav-link" href="#">Home</a>
     </li>
    <li class="nav-item">
       <a class="nav-link" href="#featureList">Features</a>
     </li>


     <li class="nav-item">
       <a class="nav-link" href="#generic_price_table">Price</a>
     </li>
     <?php
     include "WebsiteSecurityFunctions.php";
     if($validSession == true) ###they are logged in
       {
         echo "<li class='nav-item'>
         <a class='nav-link' href='./AccountOptions.php'>Account Options</a>
         </li>
         ";
         echo "<li class='nav-item'>
         <a class='nav-link' href='./logout.php'>Logout</a>
       </li>";

       }
     else
       {
         echo "<li class='nav-item'>
         <a class='nav-link' href='sign_up.php'>Sign Up</a>
       </li>
         <li class='nav-item'>
           <a class='nav-link' href='sign_in.php'>Login</a>
```

```
        </li>";
       }
      ?>

    </ul>
  </div>
</nav>



</section>
</nav>
<!-- END NAV BAR !-->
<!------------------------------------------------------------>




<!------------------------------------------------------------>


<section class="value-prop">
<div class="jumbotron jumbotron-top">
  <div class="headings">
    <h1>Mappa Takes Passwords Like These</h1> <!--Mappa take passwords like these -->
  </div>
  <div class="Mock_password">
    <i class="fas fa-lock-open"></i>

    <h3 class="typewrite" data-period="2000" data-
type='[ "password", "123456", "qwerty", "abc123", "rockyou", "bluedog", "000000"]'>
        <span class="wrap"></span>
    </h3>
  </div>
    <div class="headings" >

      <h1>And Makes Them Secure </h1> <!--And makes them secure.. <i class="fas fa-lock fa-
fw" style="font-size: 35;position: absolute ;
    padding-top: 7px;"></i>.. -->
      <br><br>


</div>
```

```
</section>
<!-------------------------------------------------------------->
<section class="product">
<div class="container padding">
<div class="row padding">
<div class="col-lg-6 product-brief">
<h1>Bad password practices are no more</h1>

<p>Mappa is an easy to use browser plugin that translates a user’s password to a completely new and highly se
cure password using a unique mapping key.</p>
</div>

<div class="col-lg-6">

<img src="./imgs/password.svg" width="100%" height="450px">
</div>

</div>
</div>

</section>
<!-------------------------------------------------------------->
<section class="table-section">
<div class="container padding" id = "featureList">
<div class="row padding">
<div class="col-lg-12 align_headers_center">
<h2>Ignore traditional password security practices</h2>
</div>
</div>


<div class="row padding">
<div class="col-lg-12">

<div class="table-container">
  <div class="tableWIV">
    <div class="table-headWIV">
      <div class="columnWIV"> </div>
      <div class="columnWIV">With Mappa</div>
      <div class="columnWIV">Without Mappa</div>

    </div>
    <div class="rowWIV">
```

```
    <div class="columnWIV light col-title">Reused Passwords</div>
    <div class="columnWIV light circle circle-wiivv circle-green"></div>
    <div class="columnWIV light circle circle-off circle-gray"></div>

   </div>
   <div class="rowWIV">
    <div class="columnWIV dark col-title">Small Passwords</div>
    <div class="columnWIV dark circle circle-wiivv circle-green"></div>
    <div class="columnWIV dark circle circle-off circle-gray"></div>

   </div>
   <div class="rowWIV">
    <div class="columnWIV light col-title">Commonly used passwords</div>
    <div class="columnWIV light circle circle-wiivv circle-green"></div>
    <div class="columnWIV light circle circle-off circle-gray"></div>

   </div>
   <div class="rowWIV">
    <div class="columnWIV dark col-title">Dictionary word passwords</div>
    <div class="columnWIV dark circle circle-wiivv circle-green"></div>
    <div class="columnWIV dark circle circle-off circle-gray"></div>

   </div>
   <div class="rowWIV">
    <div class="columnWIV light col-title">The password you want to use</div>
    <div class="columnWIV light circle circle-wiivv circle-green"></div>
    <div class="columnWIV light circle circle-off circle-gray"></div>

   </div>

  </div><!-- table -->
</div>
</div>

</div>









</div>

</section>
```

```
<!------------------------------------------------------------>
<br><br>
<section class="alt_pass_manager">
<br><br>

<div class="container padding">


  <div class="row padding">
  <div class="col-lg-12 align_headers_center">
<h1>A real alternative to password managers</h1>
<hr class="light">
  </div>
  </div>

<div class="row padding">
     <div class="col-lg-6">
     <img src="./imgs/maths.svg" width="100%" height="450px">
     </div>
     <div class="col-lg-6">
         <h2>The Math Checks Out</h2>
         <p>Password strength comes from the amount of guesses a hacker would have to make to brute force th
e password, i.e. guess every possible password.
         <br>The amount of guesses needed is determined by the password length and the size of the character se
t used, i.e. A-Z, a-z, 0-9 is a 62 character set.
         <br>Password strength also increases if you use a non-
dictionary word and not a password used by many people. For example, one of the most common password us
ed is "password",
         and so this is usually the first password hackers guess.
         <br>Hackers are attacking online services, all the time!
           </p>

     </div>

     <!--  !-->
  <div class="row padding break-how-it-works">
    <div class="col-lg-6">
     <h2>How it works</h2>
     <p>Mappa drastically increases the number of passwords a hacker must guess in order to brute force a syst
em. If we chose the password "lemons", Mappa would then transform this password
           according to a map that is assigned and unique for every online account a user has. <br>In this case, le
mons is suddenly a secure password as it is no longer the actual password!
```

```
      Mappa can still protect you if you accidentally leave your Mappa account logged in on a shared computer,
 unlike regular password managers.</p>
    </div>
    <div class="col-lg-6">
      <iframe width="98%" height="315" src="https://www.youtube.com/embed/M8WlzL_fbhs" frameborder="
0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-
picture" allowfullscreen></iframe>

    </div>

</div>



<br><br>
<!-- WAVE
<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 1440 320">
  <path fill="#fff" fill-
opacity="1" d="M0,224L48,208C96,192,192,160,288,122.7C384,85,480,43,576,32C672,21,768,43,864,96C9
60,149,1056,235,1152,256C1248,277,1344,235,1392,213.3L1440,192L1440,320L1392,320C1344,320,1248,3
20,1152,320C1056,320,960,320,864,320C768,320,672,320,576,320C480,320,384,320,288,320C192,320,96,3
20,48,320L0,320Z"></path>
</svg>  !-->
</section>


<br><br>
<!-- TRYING PRICING SECTION FROM CODEPEN !-->

<div id="generic_price_table">
<section>
      <div class="container">
        <div class="row">
          <div class="col-md-12">
            <!--PRICE HEADING START-->
            <div class="price-heading clearfix">
              <h1>Pricing</h1>
            </div>
            <!--//PRICE HEADING END-->
          </div>
        </div>
      </div>

      <div class="container">
      <div class="row">
```

```
<!-- business pricing content start !-->
<div class="col-lg-4">

    <!--PRICE CONTENT START-->
      <div class="generic_content clearfix">

        <!--HEAD PRICE DETAIL START-->
        <div class="generic_head_price clearfix">

          <!--HEAD CONTENT START-->
          <div class="generic_head_content clearfix">

            <!--HEAD START-->
              <div class="head_bg"></div>
              <div class="head">
                  <span>Business</span>
              </div>
              <!--//HEAD END-->

          </div>
          <!--//HEAD CONTENT END-->

          <!--PRICE START-->
          <div class="generic_price_tag clearfix">
            <span class="price">
              <span class="sign">€</span>
              <span class="currency">149</span>
              <span class="cent">.99</span>
              <span class="month">/Month</span>
            </span>
          </div>
          <!--//PRICE END-->

        </div>
        <!--//HEAD PRICE DETAIL END-->

        <!--FEATURE LIST START-->
        <div class="generic_feature_list">
          <ul>
          <li><span>50 User Accounts</span> </li>
            <li><span>Unlimited Devices</span> </li>
            <li><span>Secures 50 Websites each </span></li>
            <li><span>Easy Plugin Installation</span></li>
          </ul>
        </div>
```

```
            <!--//FEATURE LIST END-->

            <!--BUTTON START-->
            <div class="generic_price_btn clearfix">
              <a class="" href="">Sign up</a>
            </div>
            <!--//BUTTON END-->

          </div>
          <!--//PRICE CONTENT END-->

        </div>
      <!-- business pricing content END !-->
        <!-- STANDARD pricing content start !-->
        <div class="col-lg-4">

          <!--PRICE CONTENT START-->
            <div class="generic_content active clearfix">

              <!--HEAD PRICE DETAIL START-->
              <div class="generic_head_price clearfix">

                <!--HEAD CONTENT START-->
                <div class="generic_head_content clearfix">

                  <!--HEAD START-->
                    <div class="head_bg"></div>
                    <div class="head">
                      <span>Standard</span>
                    </div>
                    <!--//HEAD END-->

                </div>
                <!--//HEAD CONTENT END-->

                <!--PRICE START-->
                <div class="generic_price_tag clearfix">
                  <span class="price">
                    <span class="sign">€</span>
                    <span class="currency">3</span>
                    <span class="cent">.99</span>
                    <span class="month">/Month</span>
                  </span>
                </div>
                <!--//PRICE END-->
```

```
          </div>
          <!--//HEAD PRICE DETAIL END-->

          <!--FEATURE LIST START-->
          <div class="generic_feature_list">
            <ul>
              <li><span>Single User Account</span> </li>
              <li><span>Unlimited Devices</span> </li>
              <li><span>Secures 50 Websites</span></li>
              <li><span>Easy Plugin Installation</span></li>

            </ul>
          </div>
          <!--//FEATURE LIST END-->

          <!--BUTTON START-->
          <div class="generic_price_btn clearfix">
            <a class="" href="./sign_up.php">Sign up</a>
          </div>
          <!--//BUTTON END-->

        </div>
        <!--//PRICE CONTENT END-->

      </div>
       <!-- STANDARD pricing content start !-->
    <!-- ENTERPRISE pricing content start !-->
      <div class="col-lg-4">

        <!--PRICE CONTENT START-->
        <div class="generic_content clearfix">

          <!--HEAD PRICE DETAIL START-->
          <div class="generic_head_price clearfix">

            <!--HEAD CONTENT START-->
            <div class="generic_head_content clearfix">

              <!--HEAD START-->
              <div class="head_bg"></div>
              <div class="head">
                <span>Enterprise</span>
              </div>
              <!--//HEAD END-->
```

```
                    </div>
                    <!--//HEAD CONTENT END-->

                    <!--PRICE START-->
                    <div class="generic_price_tag clearfix">
                      <span class="price">
                        <span class="currency">Custom</span>
                      </span>
                    </div>
                    <!--//PRICE END-->

                  </div>
                  <!--//HEAD PRICE DETAIL END-->

                  <!--FEATURE LIST START-->
                  <div class="generic_feature_list">
                    <ul>
                    <li><span>Unlimited User Accounts</span> </li>
                      <li><span>Unlimited Devices</span> </li>
                      <li><span>Secures Unlimited Websites</span></li>
                      <li><span>Easy Plugin Installation</span></li>
                    </ul>
                  </div>
                  <!--//FEATURE LIST END-->

                  <!--BUTTON START-->
                  <div class="generic_price_btn clearfix">
                    <a class="" href="">Sign up</a>
                  </div>
                  <!--//BUTTON END-->

                </div>
                <!--//PRICE CONTENT END-->

              </div>
            <!-- ENTERPRISE pricing content END !-->
          </div>
          <!--//BLOCK ROW END-->

      </div>
    </div>
    </section>

</div>
```

```
<!-- END DEBUGGING PRICING SECTION !-->

<footer>


<div class="container-fluid padding footerdiv">
<div class="row text-center">
  <div class="col-md-4">
  <hr class="light">
  <h5>Pages</h5>
  <hr class="light">
  <div class="footertext">
  <a href="home.php"><p>Home</p></a>
  <a href="sign_up.php"><p>Sign Up</p></a>
  <a href="sign_in.php"><p>Login</p></a>
  </div>
  </div>
  <div class="col-md-4">
  <hr class="light">
  <h5>Terms</h5>
  <hr class="light">
  <div class="footertext">
  <a href="home.php"><p>Terms and Conditions</p></a>
  <a href="home.php"><p>Privacy Policy</p></a>
  <a href="home.php"><p>Cookie Policy</p></a>
  </div>
  </div>
  <div class="col-md-4">
  <hr class="light">
  <h5>Contact</h5>
  <hr class="light">
  <div class="footertext">
  <a href="home.php"><p>mappa.ie.info@gmail.com</p></a>
  </div>
  </div>
```

```
 </div>


 </div>

 </footer>

</body>



</html>
```

Sign_in.php

Below is the code for the sign in page and functionality.

```
<!DOCTYPE html>
<html lang='en'><head><meta http-equiv='Content-Type' content='text/html; charset=UTF-8'>

   <meta name='viewport' content='width=device-width, initial-scale=1, shrink-to-fit=no'>
   <meta name='description' content=''>
   <meta name='author' content=''>


   <title>Mappa</title>



   <!-- Bootstrap core CSS -->
   <link href='./Styles/bootstrap.min.css' rel='stylesheet'>

   <!-- Custom styles for this template -->
   <link href='./Styles/signin.css' rel='stylesheet'>

 </head>
<style>
   .btn-primary {
   color: #ffffff !important;
   background-color: transparent !important;
   border: solid #fefffe !important;
}
   .btn-primary:hover{
      color:white !important;
      background-color: #38ef7d !important;
```

```
      border: solid #ffffff !important;
      outline: #38ef7d !important
   }
</style>


<?php
include "WebsiteSecurityFunctions.php";
 if($authCookieSet == true)
 {

   if(CheckAuthenticatedSession($passedSessionID) == false || $badAuthCookie == true) #invalid session
      {
         echo "<h1>INVALID SESSION - BLOCKING USER AGENT." ;



      }
   else #valid session ID,
      {
         echo "<h1>You are already logged in.<br><a class='nav-link' href='home.php'>Back to home</a>";
         echo "<script>window.location.href = './home.php'</script>"; #redirect to home
      }

 }
 else #user is not logged in
   {
      if($allowedLogin === true) #ensure user is not currently blocked from logging in.
      {
         $validDeviceCSRF = GetDeviceCSRF();
         if(isset($_POST['deviceCSRF']) && $validDeviceCSRF != "0" && $validDeviceCSRF == ($_POST['
deviceCSRF']))
         {
            UpdateDeviceCSRF() ;###CSRF PASSED WAS CORRECT, UPDATE IT.
            if(isset($_POST['username']) && isset($_POST['password'])) #user is attemtpting login.
               {
                  $badLogin = true;
                  #check username exists
                  $sanitizedUsername = Sanitize($_POST['username']);
                  $hashedPassword = ($_POST['password']);
                  if(checkUsernameExists($sanitizedUsername) == true)
                     {
                        $salt = getSaltFromUsername($sanitizedUsername);
                        if($salt != false) #a salt was returned
                        {
                           $hashedPassword = PasswordHash($hashedPassword, $salt) ;
```

```
                    #check Password Matches Username
                    if(validateCredentials($sanitizedUsername,$hashedPassword) == true)
                        {

                            echo "ISSUE-ING SESSION ID<br>You have been logged in.<br>
                            <a class='nav-link' href='home.php'>Back to home</a>";
                            #issue session ID
                            createSession($sanitizedUsername) ;
                            $badLogin = false;
                            logEvent("SUCCESSFUL LOGIN", "Logged in: $sanitizedUsername");
                            echo "<script>window.location.href = './accountOptions.php'</script>"; #redirect t
o account options


                        }
                    else #invalid login attempt..
                        {
                            echo "INVALID LOGIN ATTEMPT - WRONG PASSWORD<br>
                            <a class='nav-link' href='sign_in.php'>Try again?</a>";
                            # code...
                        }

                }

            }
        else  #username does not exist
            {
                ECHO "INVALID LOGIN ATTEMPT - USERNAME DOES NOT EXIST<br>
                <a class='nav-link' href='sign_in.php'>Try again?</a>";
            }

        if($badLogin)
        {
            $toleratedBadAttempts = 3 ;
            $badAttempts = incrementInvalidLogin();
            if($badAttempts >= $toleratedBadAttempts)
                {
                    echo "BLOCKING USER SIGNIN DUE TO $badAttempts bad logins.";
                    blockUserSignIn();
                    logEvent("Too many invalid login attempts for $sanitizedUsername", "Blocked");
                }
            else
                {
                    logEvent("Invalid login attempt for $sanitizedUsername", "Invalid attempts incremented
");
```

```
                    }
                }
            }
        }

            else #no fields set OR BAD CSRF, print login form
            {
               $deviceCSRF = GetDeviceCSRF() ;
            echo" <body class='text-center'>
               <form class='form-signin' method='POST'>
               <img class='mb-4' src='./imgs/mappa_logo.svg' alt='' width='150' height='75'>

               <label for='inputUsername' class='sr-only'>Email address</label>
               <input type='text' id='inputUsername' name = 'username' class='form-
control' placeholder='Email address' required='' autofocus=''>
               <label for='inputPassword' class='sr-only'>Password</label>
               <input type='password' id='inputPassword' class='form-
control' name = 'password' placeholder='Password' required=''>
               <input type='hidden' id='deviceCSRF' class='form-
control' name = 'deviceCSRF' value = '$deviceCSRF' required=''>
               <div class='checkbox mb-3'>
                  <label>
                  <input type='checkbox' value='remember-me'> Remember me
                  </label>
               </div>
               <button class='btn btn-lg btn-primary btn-block' type='submit'>Sign in</button>
               <p class='mt-5 mb-3 text-muted'>© 2019-2020</p>
               </form>


            </body>
            </html>";
            }

      }
      else
         {
            echo "You cannot log in for $allowedLogin seconds" ;
         }

      }
   ?>
```

Sign_up.php

This is the code and functionality for the signup page.

```
<html>
<head>
<link rel='stylesheet' href='https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css' integrity='
sha384-
Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh' crossorigin='anonymo
us'>
<script src='https://code.jquery.com/jquery-3.4.1.slim.min.js' integrity='sha384-
J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n' crossorigin='anonymou
s'></script>
<script src='https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js' integrity='sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo' crossorigin='anonymous'
></script>
<script src='https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js' integrity='sha384-
wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6' crossorigin='anonymous
'></script>
<script src='./Scripts/Sign_up_js.js'></script>
<script src='https://ajax.aspnetcdn.com/ajax/jquery.validate/1.9/jquery.validate.js' ></script>
<!------ Include the above in your HEAD tag ---------->

<link rel='stylesheet' href='./Styles/Sign_up_style.css'>
<link rel='stylesheet' href='https://cdn.lineicons.com/1.0.0/LineIcons.min.css'>
<link href='https://fonts.googleapis.com/css?family=Poppins' rel='stylesheet'>
<link rel='stylesheet' href='https://cdnjs.cloudflare.com/ajax/libs/animate.css/3.7.0/animate.css' />
<style>

.form-group
      {
         color: #000000;
      }
.container-fluid.bg {
   margin-top: 40px;
}
   </style>
</head>
<body>
<?php
include "WebsiteSecurityFunctions.php";
function checkSignUpParams()
{
```

```php
$passedSetCheck = true ;
$passedSafeCheck = true ;
$report = "";
$set = "";
$notset = "";
    if(isset($_POST['username']))
        {
            $usernameSanitized = Sanitize($_POST['username']);
            if($usernameSanitized != $_POST['username'])
            {
                $passedSafeCheck = false ;
            }
            $set = $set . "Username,";
        }
    else
        {
            $notset = $notset . "Username,";
            $passedSetCheck = false;
        }
    if(isset($_POST['email']))
        {
            $set = $set .  "Email,";
            $emailSanitized = Sanitize($_POST['email']);
            if($emailSanitized != $_POST['email'])
            {
                $passedSafeCheck = false ;
            }
        }
    else
        {
            $notset = $notset .  "Email,";
            $passedSetCheck = false;
        }
    if(isset($_POST['password']))
        {
            $set = $set .  "Password,";
            $passwordSanitized = Sanitize($_POST['password']);
            if($passwordSanitized != $_POST['password'])
            {
                $passedSafeCheck = false ;
            }
        }
    else
        {
            $notset = $notset .  "Password,";
```

```
            $passedSetCheck = false;
        }
    if(isset($_POST['passwordConfirmation']))
        {
            $set = $set .  "Password Confirmation,";
                $passwordConfirmationSanitized = Sanitize($_POST['passwordConfirmation']);
                if($passwordConfirmationSanitized != $_POST['passwordConfirmation'])
                {
                    $passedSafeCheck = false ;
                }
        }
    else
        {
            $notset = $notset .  "Password Confirmation,";
            $passedSetCheck = false;
        }
  $report = "Set: " . $set . "<br>Not Set: " . $notset ;
  return $passedSetCheck ;
}

function CheckPasswordMinLength($password)
{
  $minPasswordLength = 15;
  if(strlen($password) >= $minPasswordLength)
  {
     return true;
  }
  else
  {
     return false ;
  }
}
$allFormParamsSet = checkSignUpParams();
###NAV BAR
?>
<!-- NAV BAR !-->
<link rel="stylesheet" href="./Styles/style.css">
<nav>
<section class="nav-bar">
<nav class="navbar navbar-expand-lg navbar-light">
 <a class="navbar-brand" href="#">
   <img src="./imgs/mappa_logo.svg">
  </a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-
controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
```

```
   <i class="fas fa-bars"></i>
  </button>
  <div class="collapse navbar-collapse" id="navbarNav">
   <ul class="navbar-nav ml-auto">
     <li class="nav-item">
       <a class="nav-link" href="./home.php">Home</a>
     </li>
      <li class="nav-item">
       <a class="nav-link" href="#">Features</a>
     </li>

      <li class="nav-item">
       <a class="nav-link" href="sign_up.php">Sign Up</a>
     </li>
     <li class="nav-item">
       <a class="nav-link" href="#">Price</a>
     </li>
     <?php
     if($validSession == true)
       {
         echo "<li class='nav-item'>
         <a class='nav-link' href='./AccountOptions.php'>Account Options</a>
         </li>";
         echo "<li class='nav-item'>
         <a class='nav-link' href='./logout.php'>Logout</a>
       </li>";

       }
     else
       {
         echo "
         <li class='nav-item'>
           <a class='nav-link' href='sign_in.php'>Login</a>
         </li>";
       }
     ?>

   </ul>
  </div>
</nav>



</section>
</nav>
```

```
<!-- END NAV BAR !-->

<?php
####print NAV BAR
#ensure user is allowed sign up another account
if($allowedSignup === true)
   {
      if($allFormParamsSet) #All params are set, try signup user
      {

            $usernameSanitized = Sanitize($_POST['username']);
            $emailSanitized = Sanitize($_POST['email']);

            $passwordConfirmation = $_POST['passwordConfirmation'];
            $password = $_POST['password'];
            #Ensure all conditions are met
            #CHECK PASSWORDS ARE SAME
            $insertUser = false;
            if($password == $passwordConfirmation && CheckPasswordMinLength($password)) #passwords a
re the same and min length
               {
                  #check if username exists
                  if(checkUsernameExists($usernameSanitized) == false) #username does not exist
                     {
                        $insertUser = true ;
                     }
                  else
                     {
                        echo "User already Exists" ;
                        $insertUser = false ;
                     }
               }
            else
               {
                  echo "PASSWORDS DO NOT MATCH, Min Password length 15" ;
                  $insertUser = false ;
               }

            if($insertUser === true )
            {
               $deviceCSRF = GetDeviceCSRF() ;

               if(isset($_POST['deviceCSRF']) && $deviceCSRF == $_POST['deviceCSRF'])
               {
                     echo "USER WILL BE CREATED.";
```

```
                    $createdUser = createNewUser($usernameSanitized, $emailSanitized, $password);
                    UpdateDeviceCSRF() ;###CSRF PASSED WAS CORRECT, UPDATE IT.
                    if($createdUser)
                    {
                        echo "USER CREATED.";
                        logEvent(" NEW USER REGISTERED", "Username: $usernameSanitized");
                        blockUserSignUp(); #set the signup time for this device into the future
                    }
                    else
                        {
                        echo "Error creating user." ;
                        }
                }
                else ##csrf did not match
                {
                    echo "CSRF DID NOT MATCH. USER NOT CREATED" ;
                    echo "PASSED: " . Sanitize($_POST['deviceCSRF']) . "<br>Correct: " . $deviceCSRF . "<br";
                }


            }
        else
        {
            echo "PLEASE MATCH ALL CRITERIA.<br>
            <a class='nav-link' href='sign_up.php'>Back to signup</a>";
        }
    }



else ##no params set, print the sign up form for the user
{
    echo "<div class='container-fluid bg'>
    <div class='container'>
        <div class='row'>
        <div class='col-md-8 '>
            <div class='row'>
                <div class='col-sm-3 col-md-2 col-lg-2'>
                    <i class='lni lni-enter' aria-hidden='true'></i>
                </div>

                <div class='col-sm-9 col-md-10 col-lg-10'>
                    <h1 class='heading'>Register</h1>
                    <p>Use the form on the right-hand side to Register for a Mappa account.</p>
                </div>
```

```
            </div>

            <div class='row'>
              <div class='col-sm-3 col-md-2 col-lg-2'>
                <i class='lni lni-user' aria-hidden='true'></i>
              </div>
              <br>
              <div class='col-sm-9 col-md-10 col-lg-10'>
                <h1 class='heading'>Install the Chrome browser plugin</h1>
                <p>Download the Mappa browser extension from the Chrome Web Store</p>
              </div>
            </div>
            <br>
            <div class='row'>
              <div class='col-sm-3 col-md-2 col-lg-2'>
                <i class='lni lni-cloud-upload' aria-hidden='true'></i>
              </div>
              <br>
              <div class='col-sm-9 col-md-10 col-lg-10'>
                <h1 class='heading'>Login</h1>
                <p>Use this website to login to Mappa, now you're ready to use Mappa!</p>
              </div>
            </div>
            <br>
            <div class='row'>
              <div class='col-sm-3 col-md-2 col-lg-2'>
                <i class='lni lni-lock' aria-hidden='true'></i>
              </div>
              <br>
              <div class='col-sm-9 col-md-10 col-lg-10'>
                <h1 class='heading'>Begin switching your passwords to mappa secured ones</h1>
                <p>Enter your login details on sites you would normally use, before hitting the login button, op
en the Mappa plugin, select the map to use and hit the big Map button! <br>You will see your password chang
e, then submit the form on the site!</p>
              </div>
            </div>
          </div>

          <div class='col-md-4'>
              <div class='card regform wow bounce animated' data-wow-delay='0.05s'>
                <div class='card-body regform'>
                <div class='myform form'>
                  <div class='logo mb-3'>
                  <div class='col-md-12 text-center form-header-signup'>
                    <h1 class='sign'>Sign Up</h1>
```

```
                    </div>
                    </div>
                    <form action='sign_up.php' name='registration' class = 'form-body-signup'method='POST'>
                    <div class='form-group'>
                       <label for='Signup_username'>Username</label>
                       <input type='text'  name='username' class='form-control' id='firstname' aria-
describedby='emailHelp' placeholder='Enter Username'>
                    </div>

                    <div class='form-group'>
                       <label for='Signup_email'>Email address</label>
                       <input type='email' name='email'  class='form-control' id='email' aria-
describedby='emailHelp' placeholder='Enter email'>
                    </div>
                    <div class='form-group'>
                       <label for='password'>Password</label>
                       <input type='password' name='password' id='password'  class='form-control' aria-
describedby='passwordHelp' placeholder='Enter Password'>
                    </div>
                    <div class='form-group'>
                       <label for='passwordConfirmation'>Confirm Password</label>
                       <input type='password' name='passwordConfirmation' id='passwordConfirmation'  class='f
orm-control' aria-describedby='passwordHelp' placeholder='Confirm Password'>
                    </div>";
                    $deviceCSRF = GetDeviceCSRF() ;
                    echo "
                    <input type='hidden' name='deviceCSRF' id='deviceCSRF' value='$deviceCSRF'>

                    <div class='col-md-12 text-center mb-3'>
                       <button type='submit' class=' submit-signup btn btn-block mybtn btn-primary tx-
tfm '>Sign Up</button>
                    </div></form>
                    <div class='col-md-12 '>
                       <div class='form-group'>
                          <p class='text-center sign-in-account-
form'><a href='sign_in.php' id='signin'>Already have an account?</a></p>
                       </div>
                    </div>
                       </div>
                 </div>
              </div>
           </div>
      </div>
      </div>
   </div>" ;
```

```
    }
}
else #not allowed sign up right now
{
    echo "You can't sign up again for $allowedSignup seconds" ;
}

if($authCookieSet == true) #check if user is logged in
  {

    if(CheckAuthenticatedSession($passedSessionID) == false || $badAuthCookie == true) #invalid session
        {
            echo "<h1>INVALID SESSION - BLOCKING USER AGENT." ;



        }
    else #valid session ID,
        {
            echo "<h1>You are already logged in.<br> <a class='nav-link' href='home.php'>Back to home</a>";
        }

  }
?>

</body>

</html>
```

AccountOptions.php

Below is the code for the account functions required by the user, such as deleting their account or a map, changing password or email address.

```
<html>



<head>
<meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="Description of mappa here"></head>
    <title>Mappa</title>
<script src="https://kit.fontawesome.com/eca0e10f93.js" crossorigin="anonymous"></script>
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrit
y="sha384-
Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonym
ous">
```

```
<script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-
J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n" crossorigin="anonymo
us"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymou
s"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js" integrity="sha384-
wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="anonymou
s"></script>
<link rel="stylesheet" href="./Styles/style.css">

<style>


  body {
      background: #11998e;  /* fallback for old browsers */
  background: -webkit-linear-gradient(to right, #38ef7d, #11998e);  /* Chrome 10-25, Safari 5.1-6 */
  background: linear-
gradient(to right, #38ef7d, #11998e); /* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+
*/

  }

  .tablet-form {

      background-color: white;
      border: 5px solid #ffffff;
      border-radius: 40px 40px 40px 40px;
      width: 40%;
      margin: auto;
      margin-top:50px;
      margin-bottom:50px;
      padding:20px;
      min-width:350px;
      min-height:500px;
      word-wrap: break-word;
  }

  h1.login-hello-tablet {
    text-align: center !important;
    color: grey;
    font-weight: 400;
}
}
```

```css
.footerDiv h5 {
  color: white;
}
.footerDiv p {
  color: white;
  font-weight: 600;
}

.login-hello-tablet{
    text-align: left !important;

}

.form-control  {

width:50%;
margin: auto;
}

.delete-option {
    color: red;
}
.align_headers_center h1, p, h3 {
    text-align: left;
}

.btn {
    width:40%;
    margin: auto;
    margin-top:10px;
}

/* List of options that can be select */
.accountOptionsSelection{
    cursor:pointer !important;;
}
/*Form heading that appears on button click */
.accountOptionsFormHeading{
color:black !important;;
}
/*Submit button for forms that appear */
.accountOptionsSubmit {
  background-color: #38ef7d !important;
  min-width: fit-content;
  border: none;
```

```
}
h2.accountOptionsSelection {
    text-align: center;
    font-size: medium;
    transition: all .2s ease-in-out;
    word-wrap: normal;
}
h2.accountOptionsSelection:hover {
    transform: scale(1.1);
}

/*Entire whitespace tablet*/
.container.padding {
    margin-bottom: 50px;
}
/*function response messages */
.response-message{

}

  </style>
</head>
<body>

<?php

include "WebsiteSecurityFunctions.php" ;

$checkMark = "✔" ;
$crossMark = "✕" ;
if($validSession == true) #user logged in
    {
        echo "<script src='./Scripts/accountOptionsScript.js'></script>";
        $functionResponseMessage = ""; #blank response message
        $currentCSRF = getCurrentCSRF($passedSessionID);
        ##deleteAccountFunctionality#########################################################################
        if(isset($_POST['deletePassword']) && isset($_POST['csrf'])) ##form fields are set
        {
            ####
            $hashedPassword = ($_POST['deletePassword']);
              ### username is in $loggedInUsername
               $salt = getSaltFromUsername($loggedInUsername);
               if($salt != false) #a salt was returned
               {
```

```
            $hashedPassword = PasswordHash($hashedPassword, $salt) ;

            #check Password Matches Username
            if(validateCredentials($loggedInUsername,$hashedPassword) == true) #password matches
               {
                  if(isset($_POST['csrf']) && $currentCSRF == Sanitize($_POST['csrf']))
                     {
                     #echo "DELETING ALL DATA IN USER ACCOUNT <br><br>
                     #<a class='nav-link' href='home.php'>Back to home</a>";
                     $functionResponseMessage = "Your account has been delighted. We hope you enjoyed M
appa!";
                     updateCSRF($passedSessionID);
                     $successDelete = destroyUserData($loggedInUsername);

                     if($successDelete)
                     {
                        #echo "DATA DELETED" ;
                        $functionResponseMessage = "Account Deleted $checkMark";
                     }
                     else
                     {
                       # echo "" ;
                        $functionResponseMessage = "ERROR DELETING DATA. PLEASE CONTACT MA
PPA. $crossMark";
                     }
                  }
                  else
                   {
                     #echo "INVALID CSRF TOKEN";
                     $functionResponseMessage = "Error $crossMark";
                   }
               }
            else #invalid login attempt..
               {
                  #echo "WRONG PASSWORD<br>
                  #<a class='nav-link' href='AccountOptions.php'>Try again?</a>";
                  $functionResponseMessage = "Wrong Password $crossMark";
                  # code...
               }

         }


      ######
   }
```

```
    ##changePassword unctionality#########################################################################
####################
    elseif(isset($_POST['changePassword']) && isset($_POST['newPassword']) && isset($_POST['confirmN
ewPassword']) && isset($_POST['csrf2']))
        {
            ####

            $hashedPassword = ($_POST['changePassword']);
            ### username is in $loggedInUsername
            $salt = getSaltFromUsername($loggedInUsername);
            if($salt != false) #a salt was returned
            {
                $hashedPassword = PasswordHash($hashedPassword, $salt) ;
                $hashedPassword1 = $_POST['newPassword'];
                $hashedPassword1 =  PasswordHash($hashedPassword1, $salt) ;
                $hashedPassword2 = $_POST['confirmNewPassword'];
                $hashedPassword2 =  PasswordHash($hashedPassword2, $salt) ;
                if($hashedPassword1 == $hashedPassword2 )
                {
                #check Password Matches Username
                if(validateCredentials($loggedInUsername,$hashedPassword) == true) #password matches
                    {
                        if(isset($_POST['csrf2']) && $currentCSRF == Sanitize($_POST['csrf2']))
                            {
                            #echo "Changing Users Password <br><br>
                            #<a class='nav-link' href='home.php'>Back to home</a>";
                            $functionResponseMessage = "Password Changed $checkMark";
                            updateCSRF($passedSessionID);
                            ##change users data here.
                            changePassword($loggedInUsername, $hashedPassword1) ;
                            destroyAllSessions($loggedInUsername) ;

                            }
                        else
                            {
                            #echo "INVALID CSRF TOKEN";
                            $functionResponseMessage = "ERROR $crossMark";
                            }
                    }
                else #invalid login attempt..
                    {
                        #echo "WRONG PASSWORD<br>
                        #<a class='nav-link' href='AccountOptions.php'>Try again?</a>";
                        $functionResponseMessage = "Wrong Password $crossMark";
                        # code...
```

```
                }
            }
            else ##new passwords do not match
                {
                    #echo "NEW PASSWORDS DO NOT MATCH" ;
                    $functionResponseMessage = "New Passwords do not match $crossMark";
                }
            }
        }
        ##changeEmail unctionality#########################################################################
        elseif(isset($_POST['newEmail']) && isset($_POST['confirmNewEmail']) && isset($_POST['confirmEmailPassword']) && isset($_POST['csrf3']) )
    {
            ####

            $hashedPassword = ($_POST['confirmEmailPassword']);
            ### username is in $loggedInUsername
            $salt = getSaltFromUsername($loggedInUsername);
            if($salt != false) #a salt was returned
            {
                $hashedPassword = PasswordHash($hashedPassword, $salt) ;
                $safeEmail = Sanitize($_POST['newEmail']);
                $safeEmailConfirm = Sanitize($_POST['confirmNewEmail']);
                if($safeEmail == $safeEmailConfirm )
                {
                #check Password Matches Username
                if(validateCredentials($loggedInUsername,$hashedPassword) == true) #password matches
                    {
                        if(isset($_POST['csrf3']) && $currentCSRF == Sanitize($_POST['csrf3']))
                            {
                            #echo "Changing Users Email <br><br>
                            # <a class='nav-link' href='home.php'>Back to home</a>";
                            $functionResponseMessage = "User's email changed $checkMark";
                            updateCSRF($passedSessionID);
                            ##change users email data here.
                            changeEmail($safeEmail, $loggedInUsername);
                            }
                        else
                            {
                            #echo "INVALID CSRF TOKEN";
```

```
                        $functionResponseMessage = "ERROR $crossMark";
                    }
                }
            else #invalid password
                {
                  # echo "WRONG PASSWORD<br><a class='nav-
link' href='AccountOptions.php'>Try again?</a>";
                    $functionResponseMessage = "Wrong Password $crossMark";
                    # code...
                }




            }
            else ##new passwords do not match
                {
                    #echo "NEW EMAILS DO NOT MATCH" ;
                    $functionResponseMessage = "Emails do not match $crossMark";
                }
        }
    }
    ##delete Map functionality#############################################################################
################
    elseif(isset($_POST['deleteMapPassword']) && isset($_POST['deleteMapName']) && isset($_POST['csr
f4']) )
    {
         ####

        $hashedPassword = ($_POST['deleteMapPassword']);
        ### username is in $loggedInUsername
        $salt = getSaltFromUsername($loggedInUsername);
        if($salt != false) #a salt was returned
        {
            $hashedPassword = PasswordHash($hashedPassword, $salt) ;
            if(validateCredentials($loggedInUsername,$hashedPassword) == true) #password matches
                {
                    if(isset($_POST['csrf4']) && $currentCSRF == Sanitize($_POST['csrf4']))
                        {
                            $sanitizedMap = Sanitize($_POST['deleteMapName']);
                            #echo "Deleting map: $sanitizedMap" ;
                            $functionResponseMessage = "Deleted map: $sanitizedMap";
                            deleteMap($loggedInUsername, $sanitizedMap);
                     }
                else
```

```
                    {
                      # echo "INVALID CSRF TOKEN";
                       $functionResponseMessage = "Error $crossMark";
                    }
                }




            else ##new passwords do not match
                {
                  # echo "invalid password" ;
                   $functionResponseMessage = "Invalid Password $crossMark";
                }
        }
    }




    # else #print the forms for users
     #{
        $currentCSRF = getCurrentCSRF($passedSessionID);
        echo"
        <!--- HEADER !-->
          <body class='text-center'><BR>


    <nav>
        <section class='nav-bar'>
        <nav class='navbar navbar-expand-lg navbar-light'>
        <a class='navbar-brand' href='./home.php'>
          <img src='./imgs/mappa_logo.svg'>
        </a>
        <button class='navbar-toggler' type='button' data-toggle='collapse' data-target='#navbarNav' aria-
controls='navbarNav' aria-expanded='false' aria-label='Toggle navigation'>
          <i class='fas fa-bars'></i>
        </button>
        <div class='collapse navbar-collapse' id='navbarNav'>
          <ul class='navbar-nav ml-auto'>
```

```
            <li class='nav-item'>
               <a class='nav-link' href='./logout.php'>Logout</a>
               </li>




            </ul>
          </div>
          </nav>

       </section>
       </nav>

          <!--- END HEADER !-->
          <div class='tablet-form'>

<h1 class='login-hello-tablet'>Hello $loggedInUsername</h1>
<hr>
<div class='container padding'>
<div class='row text-center padding'>
  <div class='col-xs-4 col-sm-4 col-md-3'>
     <h2  id = 'deleteMapOption' class='accountOptionsSelection'><i class='fas fa-
minus' ></i><br>Delete Map</h2>
  </div>
  <div class='col-xs-4 col-sm-4 col-md-3'>
     <h2 id = 'changeEmailOption' class='accountOptionsSelection'><i class='far fa-
envelope' ></i><br>Change Email</h2>
  </div>
  <div class='col-xs-4 col-sm-4 col-md-3'>
     <h2 id='changePasswordOption' class='accountOptionsSelection' ><i class='fas fa-exchange-
alt' ></i><br>Change Password</h2>
  </div>
  <div class='col-xs-12 col-sm-12 col-md-3'>
     <h2  id = 'deleteAccountOption' class='accountOptionsSelection delete-option'><i class='fas fa-
times' ></i><br>Delete Account</h2>
  </div>
</div>
<hr>
</div>
```

```
 <!-- DELETE ACCOUNT FORM ----------------------------------------------------------------------------------------
----!-->

   <div id = 'deleteAccountForm'>


   <form action = './AccountOptions.php' class='form-signin' method='POST'>




     <input type='password' id='inputPassword' class='form-
control' name = 'deletePassword' placeholder='Password' required=''>
     <input type='hidden' name='csrf' id='csrf' value='$currentCSRF'>

     <button class='btn btn-lg btn-primary btn-
block accountOptionsSubmit' type='submit'>Delete Account</button>

     </form>
   </div>
     <!-- CHANGE PASSWORD FORM ---------------------------------------------------------------------------------
-----------!-->
     <div id = 'changePasswordForm'>

     <form action = './AccountOptions.php' class='form-signin' method='POST'>




     <input type='password' id='changePassword' class='form-
control' name = 'changePassword' placeholder='Current Password' required=''>
    <br>
     <input type='password' id='newPassword' class='form-
control' name = 'newPassword' placeholder='New Password' required=''>
     <br>
     <input type='password' id='confirmNewPassword' class='form-
control' name = 'confirmNewPassword' placeholder='Confirm Password' required=''>
     <input type='hidden' name='csrf2' id='csrf2' value='$currentCSRF'>

     <button class='btn btn-lg btn-primary btn-
block accountOptionsSubmit' type='submit'>Change Password</button>

     </form>
   </div>
```

```
   <!-- DELETE MAP FORM ----------------------------------------------------------------------------------------
!-->
   <div id = 'deleteMapForm'>

   <form action = './AccountOptions.php' class='form-signin' method='POST' id='deleteMapForm'>




   <input type='password' id='deleteMapPassword' class='form-
control' name = 'deleteMapPassword' placeholder='Current Password' required=''>
  <br>

  <select name='list_of_maps' class = 'map-list' id = 'list_of_maps' size='number_of_options'>
  <option value='Map_1' id='Map_1'>Initial Val</option>
  </select>
   <input type='hidden' id='deleteMapName' class='form-control' name = 'deleteMapName'  required=''>
   <input type='hidden' name='csrf4' id='csrf4' value='$currentCSRF'>

   <button class='btn btn-lg btn-primary btn-block accountOptionsSubmit' type='submit'>Delete Map</button>

   </form>
 </div>
    <!-- UPDATE EMAIL FORM ----------------------------------------------------------------------------------
-----!-->
    <div id = 'changeEmailForm'>


    <form action = './AccountOptions.php' class='form-signin' method='POST'>




    <input type='email' id='newEmail' class='form-
control' name = 'newEmail' placeholder='New Email Address' required=''>
    <br>
    <input type='email' id='confirmNewEmail' class='form-
control' name = 'confirmNewEmail' placeholder='Confirm Email' required=''>
    <br>
    <input type='password' id='confirmEmailPassword' class='form-
control' name = 'confirmEmailPassword' placeholder='Password' required=''>
    <input type='hidden' name='csrf3' id='csrf3' value='$currentCSRF'>

    <button class='btn btn-lg btn-primary btn-
block accountOptionsSubmit' type='submit'>Change Email</button>
```

```
    </form>
   </div>
<div class='response-message' id='response-message'>$functionResponseMessage
</div>
</div>
        <!-- END TABLET FORM !-->



        ";
     # }
   }
else
   {
      Echo "You are not not logged in.<br><a href='./sign_in.php'> Login Now</a>
      ";
   }
?>
   </body>
        </html>
```

GenerateScript.php

This code allows map names and corresponding values to be retrieved by the browser extension.

```
<?php
include "WebsiteSecurityFunctions.php";
#header("Access-Control-Allow-Origin:" . $_SERVER['HTTP_ORIGIN']);
#header("Access-Control-Allow-Credentials: true");

   function getMapFromDomain($requestedMap,$username)
      {

         $retrievedMap= "0";
            $conn = connectToDB(true); #connect db




               #update sessionID in DB, set to "-" (not logged in)
               $stmt = $conn->prepare("select map from map WHERE domain = ? AND username = ?");
               $stmt->bind_param("ss", $requestedMap,$username);
               $stmt->execute();
               $result = $stmt->get_result();
            while($row = $result->fetch_assoc())
```

```
            {
                $retrievedMap = $row['map'];
            }

        connectToDB(false); #disconnect db
        return $retrievedMap;
    }
    ##get users maps
    function getUsersMaps($username)
    {

        $retrievedMap = "0";
        $retrievedDomain = "0";
        $usersDetails = null;
            $conn = connectToDB(true); #connect db
                #update sessionID in DB, set to "-" (not logged in)
                $stmt = $conn->prepare("select * from map where username = ?");
                $stmt->bind_param("s", $username);
                $stmt->execute();
                $result = $stmt->get_result();
                $count = 1 ;
            while($row = $result->fetch_assoc())
            {

                #echo "*COUNT$count*" ;
                $retrievedMap = $row['map'];
                $retrievedDomain = $row['domain'];
                $usersDetails .= "$retrievedDomain-";
                $count++;
            }

        connectToDB(false); #disconnect db
        return $usersDetails;
    }


function getTldAndDomain($origin)
  {
  #echo "Origin: " . $_SERVER['HTTP_ORIGIN'];

            // you can add more to it if you want
  $urlMap = array('com', 'co.uk', 'ie', 'org', 'net');

  $host = "";
```

```
   #$origin = $_SERVER['HTTP_ORIGIN'];
   $urlData = parse_url($origin);
   $hostData = explode('.', $urlData['host']);
   $hostData = array_reverse($hostData);

   if(array_search($hostData[1] . '.' . $hostData[0], $urlMap) !== FALSE) #use false because it returns the posi
tion if found.
      {
         $host = $hostData[2] . '.' . $hostData[1] . '.' . $hostData[0];
      }
      elseif(array_search($hostData[0], $urlMap) !== FALSE)
      {
         $host = $hostData[1] . '.' . $hostData[0];
      }


   return $host ;  #this is the registered domain name with no sub domains.
   }

###end functions

   if(isset($_POST['user']))
   {
      $requestedMap = Sanitize($_POST['user']); ##ensure passed map is safe..

            if($authCookieSet == true)
               {

                  if(CheckAuthenticatedSession($passedSessionID) == false || $badAuthCookie == true) #invalid
 session
                     {
                        echo "<h1>INVALID SESSION - BLOCKING USER AGENT." ;
                     }

                  else #valid session ID,
                     {
                        $username = getUsernamefromSessionID($passedSessionID);
                        if(isset($_POST['retrieve'])) ##user needs map
                           {
                              $usersMaps = getUsersMaps($username);
                              #respond with the users map
                              echo $usersMaps;
                           }
                        else
                           {
```

```
                        #$origin = getTldAndDomain($_SERVER['HTTP_ORIGIN']);
                        $map = getMapFromDomain($requestedMap,$username) ;
                        #echo "ENTERED VALID SESSION" ;
                        if(strlen($map) > 62)
                           {
                              echo "$map" ;
                           }
                        else #no map exists for this domain...
                           {

                              echo "NO MAPFILE EXISTS FOR USER:";
                           }
                     }
                  }
               }


            else
            {
               echo "COOKIE NOT SET" ;
            }

   }
   else
   {
      echo "_POST-USER-  NOT SET" ;
   }


   #echo "COOKIE SENT" ;




?>
```

genMap.php

This page contains the functionality for generating maps.

```
<?php

include "WebsiteSecurityFunctions.php";
```

```
##PAGE FUNCTIONS

###PAGE LOGIC
if($authCookieSet == true)
   {

      if(CheckAuthenticatedSession($passedSessionID) == false || $badAuthCookie == true) #invalid session
         {
            echo "<h1>INVALID SESSION - BLOCKING USER AGENT." ;
         }

      else #valid session ID,
         {

         ###MAP GENERATION#######
         #gen map, ensure constString and charString are the same...
         $constString = "QWERTYUIOPASDFGHJKLZXCVBNMqwertyuiopasdfghjklzxcvbnm1234567890!
<>._#@[]{}()+="; #alphanumeric for proof of concept
         $charString = "QWERTYUIOPASDFGHJKLZXCVBNMqwertyuiopasdfghjklzxcvbnm1234567890!<
>._#@[]{}()+="; #alphanumeric for proof of concept

         $mapString = "";
         $count = 0 ;
         while(strlen($charString) != 0)
         {
             ##jumble chars...
            $maxRand = strlen($charString);
            $charLocation = random_int( 1, $maxRand );
            if($charLocation == strlen($charString)) #handling requirements of random_int function
               {
                     $charLocation-- ;
               }
            #echo "<br>CHAR-STRING LEN " . $maxRand . "<br> RandGen: $charLocation" ;
            $mapString .= $charString{$charLocation}  ;

         if(strlen($charString) > 1)
         {
            #charRemoved was first
            if($charLocation == 0)
               {
               #$charString = $charString.Substring(1) ;
               $charString = substr($charString, 1);
               }
            #charRemoved was last
            elseif($charLocation == $maxRand-1)
```

```
            {
               #$charString = $charString.Substring(0, $($maxRand-1));
               $charString = substr($charString, 0,$maxRand-1 );
            }
         #char removed was in middle
         else
            {
            # $charString = $charString.Substring($charLocation+1) + $charString.Substring(0, $charLocatio
n-1);
               $charString = substr($charString, 0,$charLocation ) .substr($charString, $charLocation+1)  ;
            }
      }
      else #last char in char string
      {
         $charString = "" ;
      }
         #decide whether to add 0-3 extra chars...

         $maxToAdd = 4 ;# 0 - 3
         $numToAdd = random_int ( 1, $maxToAdd-1 );

         #do the adding
         $i = 0;
         while($i < $numToAdd)
         {
            $strSize =  strlen($constString);
            $charLocation = random_int ( 1, $strSize-1 );
            $mapString .= $constString[$charLocation] ;
            $i++ ;
         }
         $charSeparator = '-';
         $count++;
         $mapString .= $charSeparator;
      }
      echo "<br>Final map: $mapString";
      echo "<br> $count Character map created." ;
      ##########################END MAP GENERATION#####################################
##########################
      $username = getUsernamefromSessionID($passedSessionID);
      echo "Username from mapgen: " . $username;
      if(isset($_POST['mapName']))
         {

            $requestedMapName = $_POST['mapName'] ;
            $replacements = 0 ;
```

```
$requestedMapName = str_replace("-","_",$requestedMapName,$replacements);

echo "MapName Passed: $requestedMapName" ;
if(strlen($requestedMapName) > 0 && strlen($requestedMapName) < 64)
{

    ##ensure the name passed is a safe string
    $safeRequestedMapName = Sanitize($requestedMapName);

    ##ensure user is not brute forcing
    ##ensure user has not used all their allotted maps
    $numberOfUsersMaps = GetNumberOfUsersMaps($username);
    $numAllowedMaps = GetMaxUserMaps($username);
    $freeMapSpace = false;
    if($numberOfUsersMaps < $numAllowedMaps)
    {
       $freeMapSpace = true;
    }

    ## ensure there is no map name already in the database with that name and user
    $mapExists = checkMapNameExists($safeRequestedMapName,$username);

    ##add the map to the database
    if($mapExists == false)
    {

       if($freeMapSpace == true)
       {

          $newMapAdded = AddNewMap($username, $safeRequestedMapName, $mapString);
          if($newMapAdded == true)
          {
             echo "MAP $requestedMapName SUCCESSFULLY ADDED TO DATABASE.." ;
             logEvent("MapCreation", "Map Created for $loggedInUsername");
          }
          else
          {
             echo "ERROR ADDING MAP TO DATABASE" ;
          }
       }
       else #user has used their allowed maps
       {
          echo "User has reached the map limit. " ;
          logEvent("MapCreation", "$loggedInUsername has reached max map limit");
       }
```

```
                    }
                    else
                    {
                        echo "MAP NAME ALREADY EXISTS...";
                    }


                }
                else #map name length was too short
                {
                    echo "Map name was too short or long." ;
                }
            }



        }
    }
    else
    {
        echo "auth cookie not set" ;
    }

?>
```

assessLogin.php

This code enables the web extension to reach out and obtain details of the current users session or lack of. It enables dynamic feedback to be given to the user for login attempts.

```
<?php
include "WebsiteSecurityFunctions.php";
#header("Access-Control-Allow-Credentials: true");
$amILoggedIn = 0;
$loginAllowedStatus = 0;
$deviceCSRF = GetDeviceCSRF();
$allowedLoginSeconds = 0;
if($allowedLogin === true)
{
    $loginAllowedStatus = 1;
}
else
{
    $allowedLoginSeconds = $allowedLogin;
}
```

```
if($validSession === true)
{
   #echo "1"; ##ECHO 1 TO SAY LOGGED IN
   $amILoggedIn = 1 ;
}
else
   {

      #echo "0-$deviceCSRF" ;
   }

###echo the JSON

$json = '{
          "loggedIn":"' . $amILoggedIn . '",
          "loginAllowedStatus":"' .$loginAllowedStatus. '",
          "secondsUntilLogin":"'.$allowedLoginSeconds .'",
          "csrf":"'. $deviceCSRF ."'
       }' ;
echo $json ;


?>
```

Logout.php

This code securely logs the user out and destroys their session.

```
<html>

<head>
<meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="description" content="Description of mappa here"></head>
  <title>Mappa</title>
<script src="https://kit.fontawesome.com/eca0e10f93.js" crossorigin="anonymous"></script>
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrit
y="sha384-
Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonym
ous">
<script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-
J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n" crossorigin="anonymo
us"></script>
```

```
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="anonymous"></script>
<link rel="stylesheet" href="./Styles/style.css">

</head>
<body>
<?php
include "WebsiteSecurityFunctions.php";


if($validSession == true)
{
  echo "DESTROYING SESSION" ;
  destroySession($passedSessionID);
  logEvent("Logout", "Logged Out: $loggedInUsername");
  echo "<script>window.location.href = './home.php'</script>"; #redirect to home
}
else
{
    echo "YOU ARE NOT LOGGERD IN" ;
    echo "<script>window.location.href = './home.php'</script>"; #redirect to home
}
?>

</body>

</html>
```

Web Server Style Code

The following code is also hosted on the Web Server and is used to enhance the interface of the website. Bootstrap, which is a css framework is also pulled in to each file.

Sign_up_style.css

These are the styles used on the sign up page.

```
html,body{
    color: #fff;
    background: #11998e;  /* fallback for old browsers */
background: -webkit-linear-gradient(to right, #38ef7d, #11998e);  /* Chrome 10-25, Safari 5.1-6 */
```

```
background: linear-
gradient(to right, #38ef7d, #11998e); /* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+
*/
margin: auto;
}
.lni{
    font-size: 64px;
}

.bg{
    background: #11998e;  /* fallback for old browsers */
background: -webkit-linear-gradient(to right, #38ef7d, #11998e);  /* Chrome 10-25, Safari 5.1-6 */
background: linear-
gradient(to right, #38ef7d, #11998e); /* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+
*/
height:100%;
}

.regform{
    box-shadow: 0px 8px 9px 0px rgba(69, 69, 69, 0.25);
}

.sign{
    color: #000;
}
.submit-signup{
    border: 1px solid #2ECC71;
    color: #2ECC71 !important;
}
.btn-primary{
    outline: #000000 !important;
    background-color: white !important;
    color: #2ecc71 !important;
    font-weight: 600;
}

.btn-primary:hover{
    outline: #000000 !important;
    background-color: #2ecc71 !important;
    color: white !important;
    border: 1px solid #2ECC71;
}

.sign-in-account-form{
    color: #2ecc71 !important;
```

```
}
.form-header-signup
{
   background-color: #2ecc71;
   color:white;
   padding: 5px
}
.sign{
   color:white !important;
}

.card-body{
   background-color:#fff;
}
```

Signin.css

These are the styles used on the sign_in.php page.

```
html,
body {
  height: 100%;
  background: #11998e;  /* fallback for old browsers */
background: -webkit-linear-gradient(to right, #38ef7d, #11998e);  /* Chrome 10-25, Safari 5.1-6 */
background: linear-
gradient(to right, #38ef7d, #11998e); /* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+
*/
}

body {
  display: -ms-flexbox;
  display: -webkit-box;
  display: flex;
  -ms-flex-align: center;
  -ms-flex-pack: center;
  -webkit-box-align: center;
  align-items: center;
  -webkit-box-pack: center;
  justify-content: center;
  padding-top: 40px;
  padding-bottom: 40px;
  background-color: #f5f5f5;
}

.form-signin {
  width: 100%;
```

```
  max-width: 330px;
  padding: 15px;
  margin: 0 auto;
  box-shadow: #38ef7d !important;
}
.form-signin .checkbox {
  font-weight: 400;
}
.form-signin .form-control {
  position: relative;
  box-sizing: border-box;
  height: auto;
  padding: 10px;
  font-size: 16px;
  box-shadow: #38ef7d !important;
}
.form-signin .form-control:focus {
  z-index: 2;
}
.form-signin input[type="email"] {
  margin-bottom: -1px;
  border-bottom-right-radius: 0;
  border-bottom-left-radius: 0;
}
.form-signin input[type="password"] {
  margin-bottom: 10px;
  border-top-left-radius: 0;
  border-top-right-radius: 0;
}
```

Style.css

These are the overarching styles throughout the website.

```
@import url(https://fonts.googleapis.com/css?family=Lato:400,100,100italic,300,300italic,400italic,700italic,
700,900italic,900);
@import url(https://fonts.googleapis.com/css?family=Raleway:400,100,200,300,500,600,700,800,900);
@import url(https://fonts.googleapis.com/css?family=Raleway:400,100,200,300,500,600,700,800,900);



.nav-bar{
```

```
position:sticky;
top:0;
z-index:10;

}
.navbar{
background: #11998e;  /* fallback for old browsers */
background: -webkit-linear-gradient(to right, #38ef7d, #11998e);  /* Chrome 10-25, Safari 5.1-6 */
background: linear-
gradient(to right, #38ef7d, #11998e); /* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+
*/
padding:0 !important;

}


.navbar-brand img {
height: 60px;
padding-left:20px;
}

.navbar-nav li {
padding:0 10px;
}

.navbar-nav li a {
  color: #fff !important;
  font-weight: bold;
  float: right;
  text-align: left;
  font-size: larger;
}
.fa-bars {
color: #fff;
font-size: 30px !important;


}

.navbar-toggler {
outline:none !important;
border:none !important;
}
```

```
.Mock_password {
  border: 1px solid #ccc!important;
  border-radius: 16px;
  background-color: #ffffff;
  width: 350px;
  min-height: 55px;
  margin: 0 auto;
  margin-top: 15px;
  margin-bottom: 15px;
}

.Mock_password i{
  padding: 12px;
  line-height: 1;
  text-align: center;
  border-right: 1px solid #cccccc;
  padding-top: 14px;
  font-size: 24px;
}




.Mock_password h3{
  display: inline !important;
    font-size: 26px;
    padding-top: 10px !important;


}

.headings {
margin: 0 auto;
text-align:center;

}



.jumbotron-top{
```

```
background: #11998e;  /* fallback for old browsers */
background: -webkit-linear-gradient(to right, #38ef7d, #11998e);  /* Chrome 10-25, Safari 5.1-6 */
background: linear-
gradient(to right, #38ef7d, #11998e); /* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+
*/
padding-bottom: 0px !important;
-webkit-box-shadow:inset 0px -10px 0px 0px #fff;
    -moz-box-shadow:inset 0px -10px 0px 0px #fff;
    box-shadow:inset 0px -10px 0px 0px #fff;
  border-radius: 0;
padding-right:0;
padding-left:0;
}
.jumbotron-top svg{
bottom:0px !important;

}

.align_headers_center h1,h2,h3{
text-align: center;
}

.tableWIV {
  border: 1px solid #eaeaea;
}

.tableWIV {
  display:table;
  font-size:16px;
  color:#222222;
  margin:10px 0;
  width: 100%;
}

.table-headWIV {
   display: table-header-group;
}

.row.padding.break-how-it-works {
  margin-right: auto !important;
  margin-left: auto !important;
}
a.navbar-brand {
  width: 100%;
}
```

```
button.navbar-toggler {
  padding-top: 15px;
  float:right;
}
.table-headWIV .columnWIV {
  font-size: 36px;
  font-weight: 700;
  border-right:1px solid #fff;
  border-bottom: 0;
}

.rowWIV {
  display:table-row; /* Defines a table row */
}

.columnWIV{
  display:table-cell; /* Defines a table cell */
  font-family: 'Poppins', sans-serif;
  padding:10px 20px;
  border-right:1px solid #fff;
  text-align: center;
  min-width: 100px !important;
}

.columnWIV.light {
  background-color: #ffffff;
}

.columnWIV.dark {
  background-color: #f9fafb;
}

.break-how-it-works {
  margin-top: 35px;
}

/* === Circles === */

.circle:after {
  content: '';
  width: 18px;
  height: 18px;
  border-radius: 50%;
  display: block;
  margin: 0 auto;
```

```
}

.circle-green:after {
  background-color: #84c441;
  border: 1px solid #84c441;
}

.circle-gray:after {
  background-color: #9b9c9b;
  border: 1px solid #9b9c9b;
}

.circle-clear:after {
  background-color: rgba(0,0,0,0);
  border: 1px solid #9b9c9b;
}

.credits {
  font-family: 'Roboto', sans-serif;
  font-size: 14px;
  font-weight: 300;
  color: #9d9d9d;
  text-align: center;
  margin: 30px 0;
}

.credits p {
  margin: 12px 0 0;
}

.credits a {
  color: #9d9d9d;
  transition: all 300ms ease-in-out;
}

.credits a:hover {
  color: #222222;
}

/* === Responsive === */

@media (max-width: 767px) {

  .table-container {
    padding: 15px;
```

```
  max-width: 500px;
  margin: 0 auto;
}


.table-headWIV {
  display: none;
}

.tableWIV, .rowWIV {
  display: block;
}

.columnWIV {
  display: none;
}

.col-title {
  display: block;
  font-size: 24px;
  font-weight: 700;
}

.circle {
  display: block;
}

.circle-wiivv:before {
  content: 'With Mappa';
  display: block;
}

.circle-off:before {
  content: "Without Mappa";
  display: block;
}



.circle:after {
  font-family: "FontAwesome";
  font-size: 30px;
  display: block;
  margin: 0 auto 18px;
  position: relative;
```

```
    top: 0;
    transform: translateY(0);
    background-color: initial;
    border: 0;
  }

  .circle-wiivv:after {
    content: "\f058";
    color: #84c441;
  }

  .circle-gray:after {
    content: "\f057";
    color: #9b9c9b;
  }

  .circle-clear:after {
    content: "\f059";
    color: #9b9c9b;
  }

}


.alt_pass_manager {
height: auto;
background: #11998e;  /* fallback for old browsers */
background: -webkit-linear-gradient(to right, #38ef7d, #11998e);  /* Chrome 10-25, Safari 5.1-6 */
background: linear-
gradient(to right, #38ef7d, #11998e); /* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+
*/
}




footer {
background: #11998e;  /* fallback for old browsers */
background: -webkit-linear-gradient(to right, #38ef7d, #11998e);  /* Chrome 10-25, Safari 5.1-6 */
background: linear-
gradient(to right, #38ef7d, #11998e); /* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+
*/
}

/* PRICING CSS */
```

```
/*PRICE COLOR CODE START*/
/*PRICE COLOR CODE START*/
#generic_price_table .generic_content{

  background-color: #f6f6f6;
}

#generic_price_table .generic_content .generic_head_price{
  background-color: #f0eded;
}

#generic_price_table .generic_content .generic_head_price .generic_head_content .head_bg{
  border-color: #e4e4e4 rgba(0, 0, 0, 0) rgba(0, 0, 0, 0) #e4e4e4;
}

#generic_price_table .generic_content .generic_head_price .generic_head_content .head span{
  color: #525252;
}

#generic_price_table .generic_content .generic_head_price .generic_price_tag .price .sign{
    color: #414141;
}

#generic_price_table .generic_content .generic_head_price .generic_price_tag .price .currency{
    color: #414141;
}

#generic_price_table .generic_content .generic_head_price .generic_price_tag .price .cent{
    color: #414141;
}

#generic_price_table .generic_content .generic_head_price .generic_price_tag .month{
    color: #414141;
}

#generic_price_table .generic_content .generic_feature_list ul li{
  color: #a7a7a7;
}

#generic_price_table .generic_content .generic_feature_list ul li span{
  color: #414141;
}
#generic_price_table .generic_content .generic_feature_list ul li:hover{
  background-color: #E4E4E4;
  border-left: 5px solid #2ECC71;
```

```
}

#generic_price_table .generic_content .generic_price_btn a{
  border: 1px solid #2ECC71;
    color: #2ECC71;
}

#generic_price_table .generic_content.active .generic_head_price .generic_head_content .head_bg,
#generic_price_table .generic_content:hover .generic_head_price .generic_head_content .head_bg{
  border-color: #2ECC71 rgba(0, 0, 0, 0) rgba(0, 0, 0, 0) #2ECC71;
  color: #fff;
}

#generic_price_table .generic_content:hover .generic_head_price .generic_head_content .head span,
#generic_price_table .generic_content.active .generic_head_price .generic_head_content .head span{
  color: #fff;
}

#generic_price_table .generic_content:hover .generic_price_btn a,
#generic_price_table .generic_content.active .generic_price_btn a{
  background-color: #2ECC71;
  color: #fff;
}
#generic_price_table{
  margin: 50px 0 50px 0;
    font-family: 'Raleway', sans-serif;
}
.row .table{
    padding: 28px 0;
}

/*PRICE BODY CODE START*/

#generic_price_table .generic_content{
  overflow: hidden;
  position: relative;
  text-align: center;
}

#generic_price_table .generic_content .generic_head_price {
  margin: 0 0 20px 0;
}

#generic_price_table .generic_content .generic_head_price .generic_head_content{
  margin: 0 0 50px 0;
```

```
}

#generic_price_table .generic_content .generic_head_price .generic_head_content .head_bg{
    border-style: solid;
    border-width: 90px 1411px 23px 399px;
  position: absolute;
}

#generic_price_table .generic_content .generic_head_price .generic_head_content .head{
  padding-top: 40px;
  position: relative;
  z-index: 1;
}

#generic_price_table .generic_content .generic_head_price .generic_head_content .head span{
    font-family: "Raleway",sans-serif;
    font-size: 28px;
    font-weight: 400;
    letter-spacing: 2px;
    margin: 0;
    padding: 0;
    text-transform: uppercase;
}

#generic_price_table .generic_content .generic_head_price .generic_price_tag{
  padding: 0 0 20px;
}

#generic_price_table .generic_content .generic_head_price .generic_price_tag .price{
  display: block;
}

#generic_price_table .generic_content .generic_head_price .generic_price_tag .price .sign{
    display: inline-block;
    font-family: "Lato",sans-serif;
    font-size: 28px;
    font-weight: 400;
    vertical-align: middle;
}

#generic_price_table .generic_content .generic_head_price .generic_price_tag .price .currency{
    font-family: "Lato",sans-serif;
    font-size: 60px;
    font-weight: 300;
    letter-spacing: -2px;
```

```
    line-height: 60px;
    padding: 0;
    vertical-align: middle;
}

#generic_price_table .generic_content .generic_head_price .generic_price_tag .price .cent{
    display: inline-block;
    font-family: "Lato",sans-serif;
    font-size: 24px;
    font-weight: 400;
    vertical-align: bottom;
}

#generic_price_table .generic_content .generic_head_price .generic_price_tag .month{
    font-family: "Lato",sans-serif;
    font-size: 18px;
    font-weight: 400;
    letter-spacing: 3px;
    vertical-align: bottom;
}

#generic_price_table .generic_content .generic_feature_list ul{
  list-style: none;
  padding: 0;
  margin: 0;
}

#generic_price_table .generic_content .generic_feature_list ul li{
  font-family: "Lato",sans-serif;
  font-size: 18px;
  padding: 15px 0;
  transition: all 0.3s ease-in-out 0s;
}
#generic_price_table .generic_content .generic_feature_list ul li:hover{
  transition: all 0.3s ease-in-out 0s;
  -moz-transition: all 0.3s ease-in-out 0s;
  -ms-transition: all 0.3s ease-in-out 0s;
  -o-transition: all 0.3s ease-in-out 0s;
  -webkit-transition: all 0.3s ease-in-out 0s;

}
#generic_price_table .generic_content .generic_feature_list ul li .fa{
  padding: 0 10px;
}
#generic_price_table .generic_content .generic_price_btn{
```

```
  margin: 20px 0 32px;
}

#generic_price_table .generic_content .generic_price_btn a{
    border-radius: 50px;
  -moz-border-radius: 50px;
  -ms-border-radius: 50px;
  -o-border-radius: 50px;
  -webkit-border-radius: 50px;
    display: inline-block;
    font-family: "Lato",sans-serif;
    font-size: 18px;
    outline: medium none;
    padding: 12px 30px;
    text-decoration: none;
    text-transform: uppercase;
}

#generic_price_table .generic_content,
#generic_price_table .generic_content:hover,
#generic_price_table .generic_content .generic_head_price .generic_head_content .head_bg,
#generic_price_table .generic_content:hover .generic_head_price .generic_head_content .head_bg,
#generic_price_table .generic_content .generic_head_price .generic_head_content .head h2,
#generic_price_table .generic_content:hover .generic_head_price .generic_head_content .head h2,
#generic_price_table .generic_content .price,
#generic_price_table .generic_content:hover .price,
#generic_price_table .generic_content .generic_price_btn a,
#generic_price_table .generic_content:hover .generic_price_btn a{
  transition: all 0.3s ease-in-out 0s;
  -moz-transition: all 0.3s ease-in-out 0s;
  -ms-transition: all 0.3s ease-in-out 0s;
  -o-transition: all 0.3s ease-in-out 0s;
  -webkit-transition: all 0.3s ease-in-out 0s;
}
@media (max-width: 320px) {
}

@media (max-width: 767px) {
  #generic_price_table .generic_content{
    margin-bottom:75px;
  }
}
@media (min-width: 768px) and (max-width: 991px) {
  #generic_price_table .col-md-3{
    float:left;
```

```
    width:50%;
  }

  #generic_price_table .col-md-4{
    float:left;
    width:50%;
  }

  #generic_price_table .generic_content{
    margin-bottom:75px;
  }
}
@media (min-width: 992px) and (max-width: 1199px) {
}
@media (min-width: 1200px) {
}
#generic_price_table_home{
   font-family: 'Raleway', sans-serif;
}

.text-center h1,
.text-center h1 a{
  color: #7885CB;
  font-size: 30px;
  font-weight: 300;
  text-decoration: none;
}
.demo-pic{
  margin: 0 auto;
}
.demo-pic:hover{
  opacity: 0.7;
}

#generic_price_table_home ul{
  margin: 0 auto;
  padding: 0;
  list-style: none;
  display: table;
}
#generic_price_table_home li{
  float: left;
}
#generic_price_table_home li + li{
  margin-left: 10px;
```

```
  padding-bottom: 10px;
}
#generic_price_table_home li a{
  display: block;
  width: 50px;
  height: 50px;
  font-size: 0px;
}
#generic_price_table_home .blue{
  background: #3498DB;
  transition: all 0.3s ease-in-out 0s;
}
#generic_price_table_home .emerald{
  background: #2ECC71;
  transition: all 0.3s ease-in-out 0s;
}
#generic_price_table_home .grey{
  background: #7F8C8D;
  transition: all 0.3s ease-in-out 0s;
}
#generic_price_table_home .midnight{
  background: #34495E;
  transition: all 0.3s ease-in-out 0s;
}
#generic_price_table_home .orange{
  background: #E67E22;
  transition: all 0.3s ease-in-out 0s;
}
#generic_price_table_home .purple{
  background: #9B59B6;
  transition: all 0.3s ease-in-out 0s;
}
#generic_price_table_home .red{
  background: #E74C3C;
  transition:all 0.3s ease-in-out 0s;
}
#generic_price_table_home .turquoise{
  background: #1ABC9C;
  transition: all 0.3s ease-in-out 0s;
}

#generic_price_table_home .blue:hover,
#generic_price_table_home .emerald:hover,
#generic_price_table_home .grey:hover,
#generic_price_table_home .midnight:hover,
```

```
#generic_price_table_home .orange:hover,
#generic_price_table_home .purple:hover,
#generic_price_table_home .red:hover,
#generic_price_table_home .turquoise:hover{
  border-bottom-left-radius: 50px;
    border-bottom-right-radius: 50px;
    border-top-left-radius: 50px;
    border-top-right-radius: 50px;
  transition: all 0.3s ease-in-out 0s;
}
#generic_price_table_home .divider{
  border-bottom: 1px solid #ddd;
  margin-bottom: 20px;
  padding: 20px;
}
#generic_price_table_home .divider span{
  width: 100%;
  display: table;
  height: 2px;
  background: #ddd;
  margin: 50px auto;
  line-height: 2px;
}
#generic_price_table_home .itemname{
  text-align: center;
  font-size: 50px ;
  padding: 50px 0 20px ;
  border-bottom: 1px solid #ddd;
  margin-bottom: 40px;
  text-decoration: none;
    font-weight: 300;
}
#generic_price_table_home .itemnametext{
    text-align: center;
    font-size: 20px;
    padding-top: 5px;
    text-transform: uppercase;
    display: inline-block;
}
#generic_price_table_home .footer{
  padding:40px 0;
}

.price-heading{
    text-align: center;
```

```
}
.price-heading h1{
  color: #666;
  margin: 0;
  padding: 0 0 50px 0;
}
.demo-button {
    background-color: #333333;
    color: #ffffff;
    display: table;
    font-size: 20px;
    margin-left: auto;
    margin-right: auto;
    margin-top: 20px;
    margin-bottom: 50px;
    outline-color: -moz-use-text-color;
    outline-style: none;
    outline-width: medium ;
    padding: 10px;
    text-align: center;
    text-transform: uppercase;
}
.bottom_btn{
  background-color: #333333;
    color: #ffffff;
    display: table;
    font-size: 28px;
    margin: 60px auto 20px;
    padding: 10px 25px;
    text-align: center;
    text-transform: uppercase;
}
.demo-button:hover{
  background-color: #666;
  color: #FFF;
  text-decoration:none;

}
.bottom_btn:hover{
  background-color: #666;
  color: #FFF;
  text-decoration:none;
}

.footerDiv h5 {
```

```
  color: white;
}
.footerDiv p {
  color: white;
  font-weight: 600;
}

/*END PRICING */
```

Web Server Scripts

The following scripts are used to enhance the functionality, usability and style of html elements and to give the user the most satisfying and profession experience using the site.

Home_typewriter_script.js

This script was taken from codepen and applied to the homepage, it gives the effect of a user typing in the bar at the top of the page, for demonstration purposes.

```
$(window).scroll(function(){
    if ($(window).scrollTop() >= 500) {
        $('nav').addClass('nav-bar');

    }
    else {
        $('nav').removeClass('nav-bar');

    }
});




var TxtType = function(el, toRotate, period) {
        this.toRotate = toRotate;
        this.el = el;
        this.loopNum = 0;
        this.period = parseInt(period, 10) || 2000;
        this.txt = '';
        this.tick();
        this.isDeleting = false;
    };

    TxtType.prototype.tick = function() {
        var i = this.loopNum % this.toRotate.length;
        var fullTxt = this.toRotate[i];
```

```
    if (this.isDeleting) {
    this.txt = fullTxt.substring(0, this.txt.length - 1);
    } else {
    this.txt = fullTxt.substring(0, this.txt.length + 1);
    }

    this.el.innerHTML = '<span class="wrap">'+this.txt+'</span>';

    var that = this;
    var delta = 200 - Math.random() * 100;

    if (this.isDeleting) { delta /= 2; }

    if (!this.isDeleting && this.txt === fullTxt) {
    delta = this.period;
    this.isDeleting = true;
    } else if (this.isDeleting && this.txt === '') {
    this.isDeleting = false;
    this.loopNum++;
    delta = 500;
    }

    setTimeout(function() {
    that.tick();
    }, delta);
};

window.onload = function() {
    var elements = document.getElementsByClassName('typewrite');
    for (var i=0; i<elements.length; i++) {
        var toRotate = elements[i].getAttribute('data-type');
        var period = elements[i].getAttribute('data-period');
        if (toRotate) {
          new TxtType(elements[i], JSON.parse(toRotate), period);
        }
    }
    // INJECT CSS
    var css = document.createElement("style");
    css.type = "text/css";
    css.innerHTML = ".typewrite > .wrap { border-right: 0.08em solid #212529}";
    document.body.appendChild(css);
};
```

accountOptionsScript.js

This script gives functionality to the buttons on the accountOptions.php page, it enables the buttons to show and hide content depending on the selected element. It also contains an ajax request to dynamically populate the map list box depending on the logged in user.

```
var deleteAccount=""
var changeEmail=""
var changePassword =""
document.addEventListener('DOMContentLoaded', function ()
{
deleteAccount = document.getElementById("deleteAccountForm")
changeEmail = document.getElementById("changeEmailForm")
changePassword = document.getElementById("changePasswordForm")
deleteMap = document.getElementById("deleteMapForm")
document.getElementById("changePasswordOption").addEventListener("click",showChangePassword)
document.getElementById('changeEmailOption').addEventListener('click', showChangeEmail);
document.getElementById('deleteAccountOption').addEventListener('click', showDeleteAccount);
document.getElementById('deleteMapOption').addEventListener('click', showDeleteMap);
var deleteMapForm = document.getElementById('deleteMapForm')


deleteMapForm.onsubmit = function(){
                    var hiddenMapNameField = document.getElementById('deleteMapName');
                    var mapListbox = document.getElementById("list_of_maps")
                    var selectedMap = mapListbox.options[mapListbox.selectedIndex].text
                    hiddenMapNameField.value = selectedMap
                    hiddenMapNameField.text = selectedMap
                    }

deleteAccount.style.display ="none"
   changeEmail.style.display ="none"
   changePassword.style.display ="none"
   deleteMap.style.display = "none"
   GetMaps()
});




//functions
function clearResponse()
{
  var responseText = document.getElementById('response-message')
  responseText.innerHTML = ""
}
function deleteMapSubmitButton()
```

```
{

}
function showDeleteAccount()
{
  clearResponse()
    deleteAccount.style.display ="block"
    changeEmail.style.display ="none"
    changePassword.style.display ="none"
    deleteMap.style.display = "none"
}
function showChangeEmail()
{
  clearResponse()
    changeEmail.style.display ="block"
    deleteAccount.style.display ="none"
    changePassword.style.display ="none"
    deleteMap.style.display = "none"
}
function showChangePassword()
{
  clearResponse()
    changePassword.style.display ="block"
    changeEmail.style.display ="none"
    deleteAccount.style.display ="none"
    deleteMap.style.display = "none"

}
function showDeleteMap()
{
  clearResponse()
    changePassword.style.display ="none"
    changeEmail.style.display ="none"
    deleteAccount.style.display ="none"
    deleteMap.style.display = "block"

}

//function to populate listbox
function GetMaps()
  {
    var Http = new XMLHttpRequest();
    var url='http://localhost/Mappa/mappa/MappaWebserver/GenerateScript.php';
    var params = "user=mappa&retrieve=yes"
```

```
  Http.withCredentials = true;

  var maps ="WAITING FOR MAP" ;
  var count = 1 ;
  Http.onreadystatechange = (e) =>
  {

    //this value is passed into the inject.js script that will be injected into the current tab, this should be the ma
p for the site.
    //console.log("Response Receieved:" + test )

        if(Http.readyState == 4) //Request is complete ; prevents from executing twice.
            {
             maps = Http.responseText
             maps = maps.substring(2) //remove random breakline
              console.log("GetMaps Server Response " + count + ":'" + maps +"'")
              count++
              var separator = "-"

             var mapSplit = maps.split(separator)

              //add to list box
              var i=0
              var mapListbox = document.getElementById("list_of_maps")
              mapListbox.innerHTML= ""
              var mapNames = [];
              ///console.log("Maps: " + maps)
              while(i < (mapSplit.length-1))
                {

                  var textVal = mapSplit[i]

                  i++

                  mapNames.push(textVal)

                }

               //sort maps alphabetically
               mapNames = mapNames.sort()
               mapNames.forEach(addToListBox);
            }

    }
    Http.open("POST", url, true);
```

```
    Http.setRequestHeader('Content-type', 'application/x-www-form-urlencoded');
  Http.send(params);
  console.log("*GET MAPS RAN*")
 }
 function addToListBox(mapName)
 {
  var mapListbox = document.getElementById("list_of_maps")
  var opt = document.createElement("option");
  opt.text = mapName;
  mapListbox.options.add(opt);
 }
```

Sign_up_js.js

This contains functionality to help the user create a valid password for the site. It does **not** provide security, as it is client side code, it is purely for usability.

```
$("#signup").click(function() {
$("#first").fadeOut("fast", function() {
$("#second").fadeIn("fast");
});
});

$("#signin").click(function() {
$("#second").fadeOut("fast", function() {
$("#first").fadeIn("fast");
});
});




$(function() {

  $("form[name='registration']").validate({
   rules: {
    username: "required",
    email: {
     required: true,
     email: true
    },
    password: {
     required: true,
     minlength: 5
    },
    passwordConfirmation: {
```

```
      required: true,
      minlength: 5
    }
  },

  messages: {
    username: "Please enter your firstname",
    password: {
      required: "Please provide a password",
      minlength: "Your password must be at least 12 characters long"
    },
    passwordConfirmation: {
      required: "Please provide a password",
      minlength: "Your password must be at least 12 characters long"
    },
    email: "Please enter a valid email address"
  },

  submitHandler: function(form) {
    form.submit();
  }
 });
});
```

Web Browser Extension Code

Below is the code utilized for the functionality and styles of the web extension, a moderate amount of this code was developed using the aid of Google Chrome Extension development documentation along with various resources such as forums and other github repositories containing similar functions that were adapted. Much of the functionality of the extension comes from interacting with php files on the server using AJAX requests.

Manifest.json

This is the manifest for the extension, specifying requirements of the extension.

```
{
  "name": "MAPPA",
  "version": "2.1.1",
  "manifest_version": 2,
  "description": "Mapping password",

  "background":
    {
      "scripts":
```

```
          [
            "background.js"
          ],
        "persistent": true
      },
 "browser_action":
      {
        "default_icon": "mappa.png",
        "default_title": "Map",
        "default_popup": "popup.html"
      },
   "permissions":
      [
        "activeTab",
        "http://*/",
        "storage"
      ]

 }
```

Inject.js

This is the script that is injected into the currently open tab once a map has been selected by the user. This code changes the values in the password input boxes according to the map passed to it by popup.js, which it is called by.

Popup.html

This is the code for the interface of the extension. This code is lengthy as it contains all the elements for every possible screen, which one is displayed is determined by popup.js.

```
<!DOCTYPE html>
<html>
  <head>

    <style>
      #loginForm {
    margin-bottom: 20px;
}
#loginErrorMessage {
    color: #e21313;
    font-weight: 600;
}
.listClass {
    display: contents;
}
.usingLabel {
```

```
    display: inline;
}
#newMapDiv {
    margin-top: 8px;
}
      body{
        background: linear-gradient(to right, #38ef7d, #11998e);
    min-height: 420px;
    max-width: 290px;
    min-width: 290px;
    border: white;
    border-style: ridge ridge;
    border-width: 6px;
    padding-right: 10px;
    padding-left: 10px;
      }
      body .refreshMaps {
    padding-right: 0px;
    background-color: transparent !important;
    border: none !important ;
    outline: none !important;
    transition: all .2s ease-in-out;
    color: white;
    margin-top: 8px;
    font-size: larger;
}
body .refreshMaps:hover {
    transform: scale(1.1);
}
  .add-map{
    font-size: 40px;
    padding-top: 16px;
    color:#FFFFFF;
    cursor:pointer;
    transition: all .2s ease-in-out;
  }
  .add-map:hover{
    color: #37ec7e;
    transform: scale(1.1)
  }

  .add-map-back{
    font-size: 40px;
    padding-top: 16px;
    color:#FFFFFF;
```

```
  cursor:pointer;
  transition: all .2s ease-in-out;

 }
.add-map-back:hover{
  color: #11998e;
  transform: scale(1.1)
 }

.header {

          min-height: 40px;

        }

    .btn-primary
      {
        outline: #000000 !important;
  background-color: grey !important;
  color: #FFFFFF !important;
  border: solid #c0c0c000 !important;
  border-width: 4px !important;
  margin-top: 20px;

      }
      .btn-secondary
      {
        outline: #000000 !important;
  background-color: grey !important;
  color: #FFFFFF !important;
  border: solid #c0c0c000 !important;
  border-width: 4px !important;
  margin-top: 20px;

      }
      .map-list {
  justify-content: right !important;
  align-self: stretch;
  border-radius: 24px;
  padding: 3px;
  margin-top: 25px;
}
    html
      {
        border: #11998e 2px;
```

```
      }
   .refreshMaps
     {
      top: 0px;
      float:right;
      background-color: #38ef7d;
      border: none;
     }
 .back-button-div
 {
   margin-bottom: 10px;
 }
 .requestedMapName-div{
   text-align: center;
 }
 .add-map-button .btn{
   text-align:center ;
   align-self: center;
   margin: 0 auto;
   width: auto !important;
   border: 1px solid #fff !important;
   color: #fff !important;
   background-color: none !important;
   font-weight: 600;
   box-shadow: none !important;

 }
 .add-map-button .btn:hover{
   background-color: #2ECC71 !important;
   color: #fff !important;
   font-weight: 600;
 }
 .add-map-button {
   margin-top: 5px;

 }
 .add-map-button .btn:active:focus {
 color: #9b7c7c;
 background-color: #161617;
 border-color: #494F57;
}
/* MAP PASSWORDS NOW BUTTON*/
.map-now-button .btn{
   text-align:center ;
```

```
    align-self: center;
    margin-left: 6px;
    margin-top: 15px;
    width: auto !important;
    border: 1px solid #fff !important;
    color: #fff !important;
    background-color: none !important;
    font-weight: 600;
    box-shadow: none !important;

  }
  .map-now-button .btn:hover{
    background-color: #2ECC71 !important;
    color: #fff !important;
    font-weight: 600;
  }
  /* MAPPA LOGIN NOW BUTTON*/
.mappa-login-button .btn{
    text-align:center ;
    align-self: center;
    border-width: 2px;
    width: 100% !important;
    border:  solid #fff !important;
    color: #fff !important;
    background-color: none !important;
    font-weight: 700;
    box-shadow: none !important;
    margin: 0 auto;
    margin-top: 20px;
    font-size: large;

  }
  .button#loginButton {
    border-width: 2px !important;
}
  .mappa-login-button .btn:hover{
    background-color: #2ECC71 !important;
    color: #fff !important;
    font-weight: 600;
  }
  /*END CSS */
    </style>
  </head>
  <script type="text/javascript" src="popup.js"></script>
  <script type="text/javascript" src="\font-awesome\js\all.js"></script>
```

```
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">
  <body>

    <!--Logo area-->
<div class="header">
  <div class = "loggedInButtonBar" id = "loggedInButtonBar">
    <button id="LogoutButton" value = "Logout" class="refreshMaps" ><i class="fas fa-sign-out-alt"></i></button>

  <button id="getMapSelection" value = "Get Maps" class="refreshMaps" style="display: none"><i class="fas fa-sync-alt"></i></button>

</div>
  <img src="./imgs/mappa_logo.svg" height= "135" width="250">
   <!-- GET CURRENT ACCOUNTS MAP LIST -->

<hr>
</div>
<!----------------------LOGGED IN OPTIONS----------------------------------->

<div id = "loggedInOptions">


    <!-- generate new maps -->


  <div id = "newMapDiv">
    <div class="back-button-div">
      <a id="createNewMap-back"><i class="fas fa-arrow-circle-left add-map-back"></i></a> <!- CREATE MAP BACK TO MAP PASSWORD BUTTON-->
    </div>
    <div class = "requestedMapName-div">
      <input type = text id="requestedMapName" value="">


    </div>
      <div class = "add-map-button">
        <button id="createNewMap_send_request" class="btn btn-outline-primary btn-block btn-sm " value = "New Map">Add New Map</button>
      </div>
  </div>
```

```
  <!-- MAP ACCOUNT BUTTON -->
  <!-- LIST BOX OF ACCOUNTS -->
    <div id="mapPasswordOptions">
      <div class="usingLabel">
        <label value="Map To Use"> </label>
      </div>
      <div class ="listClass">
        <select name="list_of_maps" class = "map-list" id = "list_of_maps" size="number_of_options">
          <option value="Map_1" id="Map_1">Initial Val</option>
          </select>
          <a id="createNewMap"><i class="fas fa-plus-circle add-map"  ></i></a>
            <div class = "map-now-button">
              <button id="mapPassword" class="btn btn-outline-primary btn-
block " value = "Map Password" >Secure Me - Map Passwords</button></button>
            </div>
    </div>
  </div>
</div>
<!--END OF LOGGED IN OPTIONS-->

<!--login error message -->
<div id = "loginErrorMessage">
  Invalid username or password
</div>
<!--END login error message -->
<div id = "loginForm">
<!-- LOgin Form !-->

    <label for='inputUsername' ></label>
    <input type='text' id='inputUsername' name = 'username' class='form-
control' placeholder='Username' required='' autofocus=''>

    <label for='inputPassword'></label>
    <input type='password' id='inputPassword' class='form-
control' name = 'password' placeholder='Password' required=''>
    <input type='hidden' id='deviceCSRF' class='form-
control' name = 'deviceCSRF' placeholder='csrf' value='' required=''>
    <div class = "mappa-login-button">
    <button class='btn btn-outline-primary btn-block ' type='' id="loginButton">Sign in</button>
   </div>

</div>
<!-- End Login Form !-->

<!-- LOgin processing screen !-->
```

```
<div id = "attemptLogin">
  ATTEMPTING LOGIN.....
</div>
<!-- end login processing screen !-->


MAPPA DEVELOPMENT
</div> </body>
</html>
```

Popup.js

This JavaScript file contains all the necessary AJAX calls and processing to enable the extension to function. **The code shown here is the development code, the ajax calls are replaced with HTTPS in the production extension and the URL used is mappa.ie, instead of localhost.**

```
//Get specific map call to server
function MapPassword() {
   var Http = new XMLHttpRequest();
   var mapListbox = document.getElementById("list_of_maps")
   var selectedMap = mapListbox.options[mapListbox.selectedIndex].text
   console.log("SELECTED MAP: " + selectedMap)
   var url='http://localhost/Mappa/mappa/MappaWebserver/GenerateScript.php'
   var params = "user=" + selectedMap

   Http.withCredentials = true;

   var test ="WAITING FOR MAP" ;
   var finished = false;
   Http.onreadystatechange = (e) =>
   {

     test = Http.responseText //this value is passed into the inject.js script that will be injected into the current ta
b, this should be the map for the site.
     console.log("Response Receieved:" + test )
         if(test.length > 62 && finished == false) //prevents from executing twice.
            {
               chrome.storage.local.set
                  ( //set the variable in local chrome storage.
                     {
                        test: test
                     },
                     function()
                     {
                        chrome.tabs.executeScript
```

```
                        (
                            {
                                file: "inject.js"
                            }
                        );
                    }
                );       ///end of setting variable in local storage
                finished = true ;
            }

    }
    Http.open("POST", url,true);
    Http.setRequestHeader('Content-type', 'application/x-www-form-urlencoded');
  Http.send(params);
  console.log("*MAP PASSWORD RAN*")
 }
 //Get list of map call to server
 function GetMaps()
 {
  var Http = new XMLHttpRequest();
  var url='http://localhost/Mappa/mappa/MappaWebserver/GenerateScript.php';
  var params = "user=mappa&retrieve=yes"

  Http.withCredentials = true;

  var maps ="WAITING FOR MAP" ;
  var count = 1 ;
  Http.onreadystatechange = (e) =>
  {

    //this value is passed into the inject.js script that will be injected into the current tab, this should be the ma
p for the site.
    //console.log("Response Receieved:" + test )

        if(Http.readyState == 4) //Request is complete ; prevents from executing twice.
            {
             maps = Http.responseText
             maps = maps.substring(2) //remove random breakline
              console.log("GetMaps Server Response " + count + ":" + maps +"")
               count++
               var separator = "-"

             var mapSplit = maps.split(separator)

               //add to list box
```

```
                var i=0
                var mapListbox = document.getElementById("list_of_maps")
                mapListbox.innerHTML= ""
                var mapNames = [];
                ///console.log("Maps: " + maps)
                while(i < (mapSplit.length-1))
                  {

                    var textVal = mapSplit[i]
                    ////var valueVal = "VALUE TEXT"
                   // var opt = document.createElement("option");
                   // opt.text = textVal;
                   // opt.value = valueVal;

                    //console.log("textVal: " + textVal)
                   // mapListbox.options.add(opt);


                    i++
                    //clear add new map field
                    var newMapField = document.getElementById("requestedMapName")
                    newMapField.value = "";
                    mapNames.push(textVal)

                  }

                 //sort maps alphabetically
                 mapNames = mapNames.sort()
                 mapNames.forEach(addToListBox);
              }

      }
      Http.open("POST", url, true);
      Http.setRequestHeader('Content-type', 'application/x-www-form-urlencoded');
   Http.send(params);
   console.log("*GET MAPS RAN*")
  }
  //add to listbox function
function addToListBox(mapName)
  {
   var mapListbox = document.getElementById("list_of_maps")
   var opt = document.createElement("option");
   opt.text = mapName;
   mapListbox.options.add(opt);
  }
```

```
//end add to listbox function
  //creating a new map
  function CreateNewMap()
  {
    var Http = new XMLHttpRequest();
    var mapName = document.getElementById("requestedMapName").value
      if(mapName.length > 5)
      {
        var url='http://localhost/Mappa/mappa/MappaWebserver/genMap.php'
        Http.withCredentials = true;

        var params = "mapName=" + mapName;
        var response ="WAITING FOR RESPONSE" ;
        var count = 1 ;
        Http.onreadystatechange = (e) =>
        {

          //this value is passed into the inject.js script that will be injected into the current tab, this should be the m
ap for the site.
          //console.log("Response Receieved:" + test )

              if(Http.readyState == 4) //Request is complete ; prevents from executing twice.
                  {
                    response = Http.responseText
                    console.log("GetMaps Server Response " + count + ":\n" + response)
                      //Reload the maps
                      setTimeout(GetMaps, 200);
                  }

          }
          Http.open("POST", url, true);
          Http.setRequestHeader('Content-type', 'application/x-www-form-urlencoded');
        Http.send(params);
        console.log("\n*CreateNewMaps MAPS RAN*")
        }
  }
//attempting login
function AttemptLogin()
  {
    //show attempting login

    requestSent = true
    console.log("TRYING LOGIN");
    var Http = new XMLHttpRequest();
    var username = document.getElementById("inputUsername").value
```

```
  var password = document.getElementById("inputPassword").value
  var csrf = document.getElementById("deviceCSRF").value

  var url='http://localhost/Mappa/mappa/MappaWebserver/sign_in.php'

  var params = "username=" + username + "&password=" +password + "&deviceCSRF=" + csrf ;
  document.getElementById("inputUsername").value = ""
  document.getElementById("inputPassword").value = ""
  Http.onreadystatechange = (e) =>
  {

        if(Http.readyState == 4) //Request is complete ; prevents from executing twice.
            {
              //response = Http.responseText

              assessLoginStatus(true) //set to true to flag it came from this function

                //Reload the maps
            }

     }
     Http.open("POST", url, true);
     Http.setRequestHeader('Content-type', 'application/x-www-form-urlencoded');
  Http.send(params);
  console.log("\n*Attempt login*")

}
function assessLoginStatus(attempted)
{

    var Http = new XMLHttpRequest();
    var url='http://localhost/Mappa/mappa/MappaWebserver/assessLogin.php'
    Http.withCredentials = true;


    var response ="WAITING FOR RESPONSE" ;
    var returnVal = false ;
    Http.open("POST", url);
     Http.setRequestHeader('Content-type', 'application/x-www-form-urlencoded');
    Http.send();
    Http.onreadystatechange = (e) =>
    {
      document.getElementById("attemptLogin").style.display = "none"; //
           var loggedIn = false
```

```
      //this value is passed into the inject.js script that will be injected into the current tab, this should be the ma
p for the site.
      //console.log("Response Receieved:" + test )

         if(Http.readyState == 4) //Request is complete ; prevents from executing twice.
             {
              try{
                response = Http.responseText
              // console.log("ASSESS LOGIN STATUS Server Response:\n"' + response +'"')
              //parse json response
                var loginStatus = JSON.parse(response); //loginStatus.loggedIn, loginAllowedStatus, secondsUnt
ilLogin,csrf
                loginStatus.loggedIn = parseInt(loginStatus.loggedIn)
                loginStatus.loginAllowedStatus = parseInt(loginStatus.loginAllowedStatus)
                loginStatus.secondsUntilLogin = parseInt(loginStatus.secondsUntilLogin)

                console.log("\nLogin Status: " + loginStatus.loggedIn + "\nAllowedLogin: " + loginStatus.login
AllowedStatus +
                            "\nSeconds Until Second :" + loginStatus.secondsUntilLogin +"\ncsrf: " + loginStatus.csr
f + "\n")
                if(loginStatus.loggedIn == 1) //successful login
                {

                  returnVal = true

                }
                else //unsuccessful login
                {

                  updateCSRF(loginStatus.csrf);
                  returnVal = false

                }
              }
             catch
             {
              console.log("ERROR REACHING MAPPA");
             }
              processLoginStatus(returnVal, attempted, loginStatus.secondsUntilLogin)
             }

      }
```

```
 }

 //check loggedIn funcion
  function processLoginStatus(loggedIn, attempted, secondsUntilLogin)
  {
    var form =  document.getElementById("loginForm");
    var loggedInOptions = document.getElementById("loggedInOptions");
    var loggedError = document.getElementById("loginErrorMessage");
    var loggedInButtons = document.getElementById("loggedInButtonBar");
    loggedError.style.display = "none";
    loggedInButtons.style.display = "none";
    if(loggedIn == true)
      {
        loggedInOptions.style.display = "block";
        form.style.display = "none";
        loggedInButtons.style.display = "block";
        console.log("LOGGED IN - DEV") ;
        setTimeout(GetMaps, 200);
        remainingAttempts = 3;
      }
    else //Not logged in
    {
      console.log("NOT LOGGED IN - DEV") ;
      loggedInOptions.style.display = "none";
      form.style.display = "block";
      if(secondsUntilLogin > 0) // user is currently blocked
      {
        loggedError.style.display = "block";
        loggedError.innerHTML = "You are blocked for another " + secondsUntilLogin + " seconds.";
        remainingAttempts = 1;
      }
      else if(attempted == true) //invalid login attempt
        {
          loggedError.style.display = "block";
          remainingAttempts--;
          if(remainingAttempts == 0)
            {
              remainingAttempts = 3;
              loggedError.innerHTML = "You have been blocked from logging in for 5 minutes."
            }
          else
            {
              loggedError.innerHTML = "Invalid username or password <br>" + remainingAttempts + " attempts r
emaining";
            }
```

```
        }
    }
  }
//logout AttemptLogout
function AttemptLogout()
{

    var Http = new XMLHttpRequest();
    var url='http://localhost/Mappa/mappa/MappaWebserver/logout.php'
    Http.withCredentials = true;

    Http.open("POST", url);
     Http.setRequestHeader('Content-type', 'application/x-www-form-urlencoded');
    Http.send();
    Http.onreadystatechange = (e) =>
    {

        if(Http.readyState == 4) //Request is complete ; prevents from executing twice.
            {

              assessLoginStatus(false) //was not a login attempt

            }
      }
}

//end logout

//function update csrf value
function updateCSRF(csrfVal)
{
 var x = document.getElementById("deviceCSRF");
 x.value = csrfVal;
}

//end function update csrf value

//handlers------------------------------------------------------------
function mapPasswordclickHandler(e) {
  setTimeout(MapPassword, 200);
}
//handlers
function getMapsclickHandler(e) {

  setTimeout(GetMaps, 500);
```

```
   main()
  }


  //handlers
//begin create map button handler
function createNewMapButtonHandler(e) {

  var mapPasswordOptions = document.getElementById("mapPasswordOptions");
  var newMapDiv = document.getElementById("newMapDiv");
  var newMapField = document.getElementById("requestedMapName")
  newMapField.value = "";
if (newMapDiv.style.display === "none") // if currently not showing
  {
    newMapDiv.style.display = "block";
    mapPasswordOptions.style.display = "none";
  }
  else  //currently showing
  {
    newMapDiv.style.display = "none";
    mapPasswordOptions.style.display = "block";
  }
}
  //end create map button handler
  function createNewMapclickHandler(e) {

    var mapPasswordOptions = document.getElementById("mapPasswordOptions");
    var newMapDiv = document.getElementById("newMapDiv");
  if (newMapDiv.style.display === "none") // if currently not showing
    {
      newMapDiv.style.display = "block";
      mapPasswordOptions.style.display = "none";
    }
    else  //currently showing
    {
      setTimeout(CreateNewMap, 200);
      newMapDiv.style.display = "none";
      mapPasswordOptions.style.display = "block";
    }
  }
  //login button click handler
  function loginButtonclickHandler(e)
    {
      document.getElementById("attemptLogin").style.display = "block";
```

```
    setTimeout(AttemptLogin, 500);

   }
  //end login button click handler
//logout click handler logoutButtonclickHandler
function logoutButtonclickHandler(e)
   {

    setTimeout(AttemptLogout, 200);

   }
  //end logout click hander
  function main() {


   //hide new map input box
   var x = document.getElementById("newMapDiv");
   x.style.display = "none";


   //show different menus depending on login status

   assessLoginStatus(false)




  }

  // Add event listeners once the DOM has fully loaded by listening for the
  // `DOMContentLoaded` event on the document, and adding your listeners to
  // specific elements when it triggers.
  document.addEventListener('DOMContentLoaded', function ()
  {
   document.getElementById('mapPassword').addEventListener('click', mapPasswordclickHandler);
   document.getElementById('getMapSelection').addEventListener('click', getMapsclickHandler);
   document.getElementById('createNewMap').addEventListener('click', createNewMapButtonHandler);
   document.getElementById('createNewMap-
back').addEventListener('click', createNewMapButtonHandler);//createNewMap-
back //createNewMap_send_request
   document.getElementById('createNewMap_send_request').addEventListener('click', createNewMapclickHa
ndler);
   document.getElementById('loginButton').addEventListener('click', loginButtonclickHandler);
   document.getElementById('LogoutButton').addEventListener('click', logoutButtonclickHandler);
```

```
  main();


});

var remainingAttempts = 3;
```