

Name : Noor Saleem id:6811

DEPARTMENT : COMPUTER SCIENCE

SUBMITTED TO : SIR SHAHAB UL ISALAM

```
<!DOCTYPE html>

<html lang="en">

<head>

<title>Bootstrap Example</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

</head>

<body>

<div class="container">

<h2>Registration form</h2>

<form action="/action_page.php">

<div class="form-group">

<label for="name">Name:</label>

<input type="name" class="form-control" id="name" placeholder="Enter name" name="name">

</div>

<div class="form-group">

<label for="email">Email:</label>
```

```
<input type="email" class="form-control" id="email" placeholder="Enter email" name="email">
</div>
<div class="form-group">
  <label for="pwd">Password:</label>
  <input type="password" class="form-control" id="pwd" placeholder="Enter password"
name="pwd">
</div>
<button type="submit" class="btn btn-default">Submit</button>
</form>
</div>
</body>
</html>
```

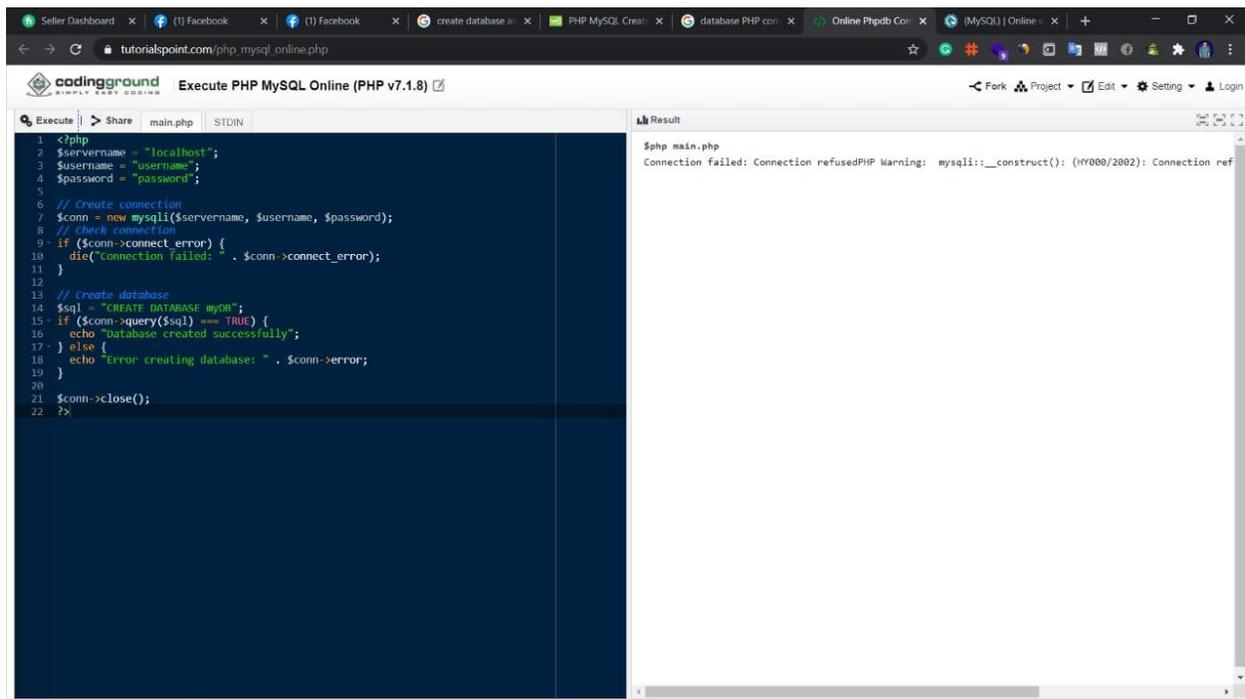
Registration form

Name:

Email:

Password:

QNO1 B.



```
1 <?php
2 $servername = "localhost";
3 $username = "username";
4 $password = "password";
5
6 // Create connection
7 $conn = new mysqli($servername, $username, $password);
8 // Check connection
9 if ($conn->connect_error) {
10     die("Connection failed: " . $conn->connect_error);
11 }
12
13 // Create database
14 $sql = "CREATE DATABASE myDB";
15 if ($conn->query($sql) !== TRUE) {
16     echo "Database created successfully";
17 } else {
18     echo "Error creating database: " . $conn->error;
19 }
20
21 $conn->close();
22 >?>
```

Result

```
$php main.php
Connection failed: Connection refusedPHP Warning: mysqli::__construct(): (HY000/2002): Connection ref
```

QNO1.C

users (

BIGINT id,

VARCHAR(?) username,

VARCHAR(?) password,

VARCHAR(254) email,

QNO.1 D

CREATE TABLE `users` (

`id` int(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,

`username` varchar(100) NOT NULL,

`email` varchar(100) NOT NULL,

```
`password` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET
<?php include('server.php') ?>
<!DOCTYPE html>
<html>
<head>
<title>Registration system PHP and MySQL</title>
<link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
<div class="header">
<h2>Register</h2>
</div>
<form method="post" action="register.php">
<?php include('errors.php'); ?>
<div class="input-group">
<label>Username</label>
<input type="text" name="username" value="<?php echo $username; ?>">
</div>
<div class="input-group">
<label>Email</label>
<input type="email" name="email" value="<?php echo $email; ?>">
</div>
<div class="input-group">
```

```
<label>Password</label>
<input type="password" name="password_1">
</div>
<div class="input-group">
  <label>Confirm password</label>
  <input type="password" name="password_2">
</div>
<div class="input-group">
  <button type="submit" class="btn" name="reg_user">Register</button>
</div>
<p>
  Already a member? <a href="login.php">Sign in</a>
</p>
</form>
</body>
</html>
```

QN.O1 F

[04:53, 9/27/2020] +92 307 9871865: <?php include('server.php') ?>

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Registration system PHP and MySQL</title>
```

```
<link rel="stylesheet" type="text/css" href="style.css">
```

```
</head>
```

```
<body>
```

```
<div class="header">
```

```
<h2>Register</h2>
```

```
</div>
```

```
<form method="post" action="register.php">
```

```
<?php include('errors.php'); ?>
```

```
<div class="input-group">
```

```
<label>Username</label>
```

```
<input type="text" name="username" value="<?php echo $username; ?>">
```

```
</div>
```

```
<div class="input-group">
```

```
<label>Email</label>
```

```
<input type="email" name="email" value="<?php echo $email; ?>">
```

```
</div>
<div class="input-group">
  <label>Password</label>
  <input type="password" name="password_1">
</div>
<div class="input-group">
  <label>Confirm password</label>
  <input type="password" name="password_2">
</div>
<div class="input-group">
  <button type="submit" class="btn" name="reg_user">Register</button>
</div>
<p>
  Already a member? <a href="login.php">Sign in</a>
</p>
</form>
</body>
</html>
```

[04:57, 9/27/2020] +92 307 9871865: <?php

```
use Google_Service_CloudKMS as Kms;
```

```
use Google_Service_CloudKMS_DecryptRequest as DecryptRequest;
```

```
use Google_Service_CloudKMS_EncryptRequest as EncryptRequest;
```

```
class KeyManager
```

```

{
    private $kms;
    private $encryptRequest;
    private $decryptRequest;
    private $projectId;
    private $locationId;
    private $keyRingId;
    private $cryptoKeyId;

    public function __construct(Kms $kms, EncryptRequest $encryptRequest, DecryptRequest
    $decryptRequest, $projectId, $locationId, $keyRingId, $cryptoKeyId)
    {
        $this->kms      = $kms;
        $this->encryptRequest = $encryptRequest;
        $this->decryptRequest = $decryptRequest;
        $this->projectId  = $projectId;
        $this->locationId  = $locationId;
        $this->keyRingId   = $keyRingId;
        $this->cryptoKeyId  = $cryptoKeyId;
    }

    public function encrypt($data)
    {
        $key      = random_bytes(SODIUM_CRYPT0_SECRETBOX_KEYBYTES);
        $nonce    = random_bytes(SODIUM_CRYPT0_SECRETBOX_NONCEBYTES);
        $ciphertext = sodium_crypto_secretbox($data, $nonce, $key);
    }
}

```



```

return [
    'data' => base64_encode($nonce . $ciphertext),
    'secret' => $this->encryptKey($key),
];
}

public function decrypt($secret, $data)
{
    $decoded = base64_decode($data);
    $key = $this->decryptSecret($secret);
    $nonce = mb_substr($decoded, 0, SODIUM_CRYPTO_SECRETBOX_NONCEBYTES, '8bit');
    $ciphertext = mb_substr($decoded, SODIUM_CRYPTO_SECRETBOX_NONCEBYTES, null, '8bit');

    return sodium_crypto_secretbox_open($ciphertext, $nonce, $key);
}

private function encryptKey($key)
{
    $this->encryptRequest->setPlaintext(base64_encode($key));

    $response = $this->kms->projects_locations_keyRings_cryptoKeys->encrypt(
        $this->getResourceName(),
        $this->encryptRequest
    );
}

```

```
    return $response['ciphertext'];  
}
```

```
private function decryptSecret($secret)
```

```
{
```

```
    $this->decryptRequest->setCiphertext($secret);
```

```
    $response = $this->kms->projects_locations_keyRings_cryptoKeys->decrypt(  
        $this->getResourceName(),  
        $this->decryptRequest
```

```
    );
```

```
    );
```

```
);
```

```
    return base64_decode($response['plaintext']);
```

```
}
```

```
private function getResourceName()
```

```
{
```

```
    return sprintf(  
        'projects/%s/locations/%s/keyRings/%s/cryptoKeys/%s',  
        $this->projectId,  
        $this->locationId,  
        $this->keyRingId,  
        $this->cryptoKeyId
```

```
    );
```

```
);
```

```
);
```

```
);
```

```
);
```

```
);
```

