Table of Contents

Introduction

This project was implemented with the front-end in PHP and back-end in MySQL. For the MySQL backend, the PHPMyAdmin tool is used to create a database, run a SQL query. I have used HTML and CSS to create web pages.

1. How did you design the server-side code and UI code?

The server-side code is designed using PHP and UI code is designed using HTML forms. The PHP script is executed on the server, the output is built on the server, and the resulting HTML pages are sent to the user's browser for rendering.

2. How can your application scale?

- The easiest way to scale web applications is by using cloud computing. The cloud service provider will manage complex IT infrastructure. There are other options available for scaling either by vertical or horizontal scaling.
- Vertical Scaling means adding more CPUs and RAM to improve the performance, but it is not cost-effective as buying more servers is expensive.
- Horizontal Scaling means adding more servers to improve the speed and performance, We need a load balancer that will redirect users to one of the application servers. The server must be stateless. PHP uses sessions to keep track of the user's state. If a user logs in on server 1 then that server knows the user is logged in. But if the load-balancer sends the next request to server 2 then this server will not know that the user is logged in. To avoid this, we can store sessions in the database. Or we can use Redis to store key-value data.

3. How do you protect your server from user input, both malicious and unintentional?

- I have used the SHA1 cryptographic hash function to encrypt the password along with salts, to create unique hashed passwords. This will generate a different hash even for the same user passwords.
- Done validation on user login and Registration so that only valid users can log in.

Implementation of Requirements

1. Reconstruct the Databases based on the given schema.
    - Create a database with the name ‘email_classification’.
    - Create tables in ‘email_classification’ database :
      Organisation
      Users
      Submissions

Screenshots are as below:
Organisation table:

#	Name	Type	Collation	Attributes	Null
1	org_id	int(11)			No
2	org_name	varchar(50)	utf8mb4_general_ci		No

Users table

#	Name	Type	Collation	Attributes	Null
1	user_id	int(11)			No
2	email	varchar(100)	utf8mb4_general_ci		No
3	username	varchar(32)	utf8mb4_general_ci		No
4	password	varchar(255)	utf8mb4_general_ci		No
5	legal_name	varchar(50)	utf8mb4_general_ci		Yes
6	organisation_id	int(11)			No

Submissions table:

#	Name	Type	Collation	Attributes	Null
1	submission_id	int(11)			No
2	submission_path	varchar(200)	utf8mb4_general_ci		No
3	class_name	varchar(50)	utf8mb4_general_ci		No
4	user_id	int(11)			No

2. SHA1 Cryptographic Hash Function.

Import the existing data and Use the SHA1 cryptographic hash function to encrypt the password along with salts, to create unique hashed passwords.

Stored encrypted passwords for new users and populated users from the user.csv file.

user_id	email	username	password
4	cmgtuv@mymailcr.com	chrismg	e3102807d0b1069299c644afb9a3315d72b9e6812a0f7e49cb...
5	upo1342@fw025.com	ulripo	e3102807d0b1069299c644afb5118a8e515f924ee96e63eede...
6	ultis2@fw025.com	ultis	e3102807d0b1069299c644afbecbdd0ef37aeaf41c44b877bc...
7	tymou135@fw025.com	tyrone3	e3102807d0b1069299c644afb875acc53eec8e96908d94a9ec...
8	cand7ysm@fw025.com	candiceM	e3102807d0b1069299c644afb4dabac3907725f83972372bb1...
9	upo1342@mymailcr.com	upo1342	e3102807d0b1069299c644afbf1e6d2af3d741916dcd6d83d5...
10	ekpe70@fw025.com	ericpeters	e3102807d0b1069299c644afb8c1f91bd2b659ffc96b242a10...
11	grancc@mymailcr.com	grant	e3102807d0b1069299c644afbbe49383160b55a3bb6d36dc4e...
12	wrets55@fw025.com	wilrets	e3102807d0b1069299c644afbfd955397bef9d6dde9c286664...
13	ytumb00@fw025.com	yymb	e3102807d0b1069299c644afbc305d542cd34dc8a438b08436...
14	stryesm@mymailcr.com	stryerM	e3102807d0b1069299c644afb986f81da4659c41389ab37642...
32	meenal@gmail.com	Meenal	eeb2c54134c0fddfc57c56d3cf4c81fc3ab04422fcf659983b...

Update SQL Query to encrypt the password populated by the user.csv file

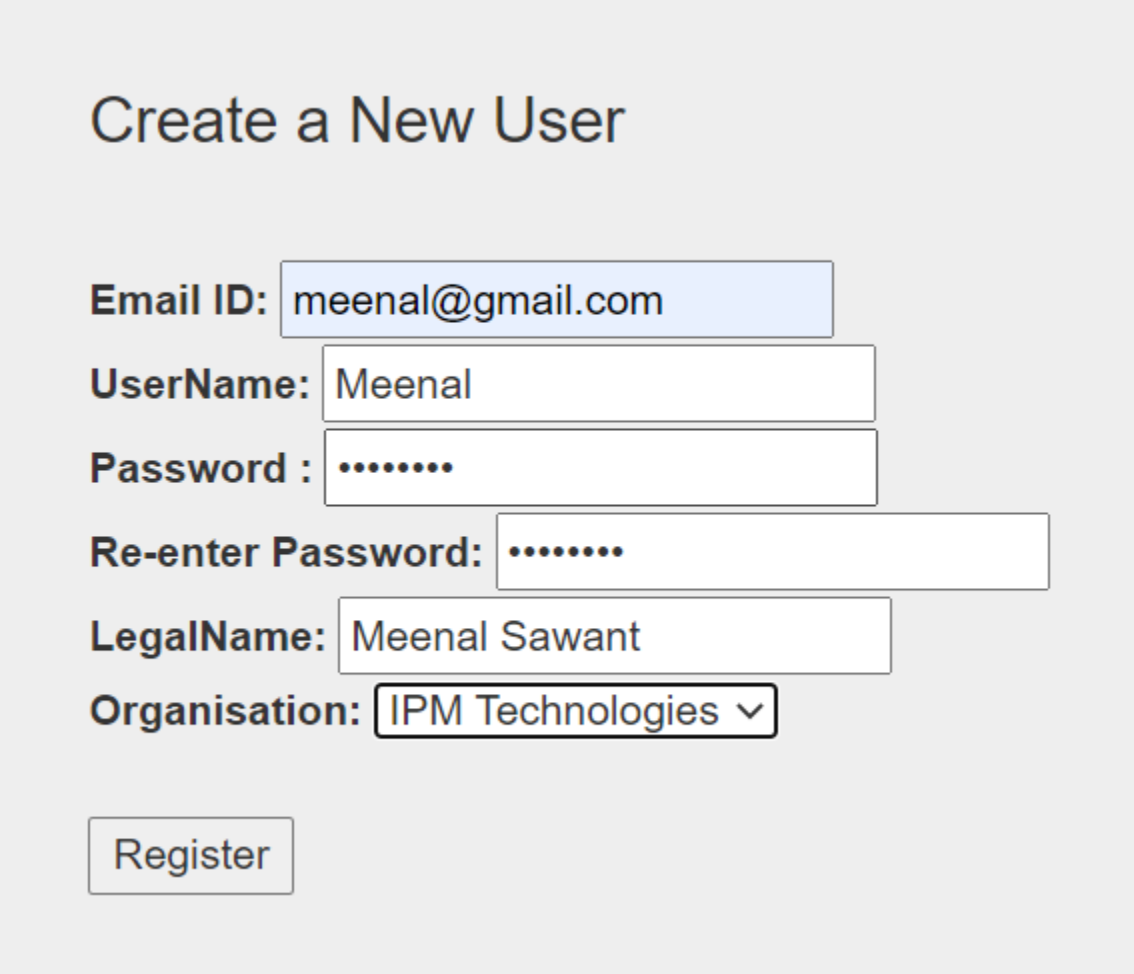
```
SET @salt = SUBSTRING(MD5(RAND()), 1, 25);
```

```
update users set password = concat(@salt, SHA1(CONCAT( @salt,
password )));


SELECT u.user_id, s.submission_id, s.submission_path,
s.class_name, u.organisation_id FROM `submissions` s
inner join users u
on u.user_id = s.user_id
WHERE organisation_id  = (select organisation_id from users
where user_id = 1)
```

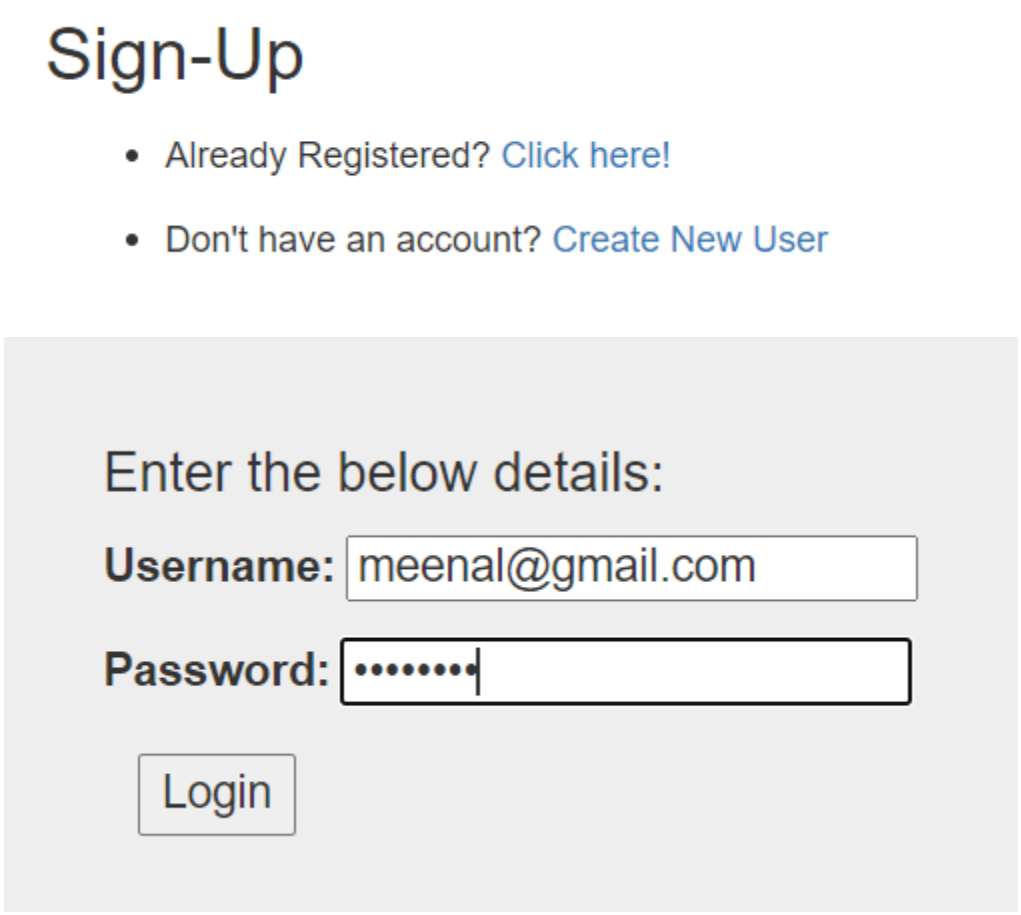
Code snippet to encrypt the password for New Users:

```php
function generateHash($plainText, $salt = NULL)
{
    if($salt == NULL) {
        $salt = substr(md5(uniqid(rand(), more_entropy: TRUE)), start: 0, length: 25);
        //$salt = substr(md5(rand(), TRUE), 0, 25);
    } else {
        //echo 'else salt';
        $salt = substr($salt, start: 0, length: 25);

    }
    //echo $salt;
    //echo '**';
    //echo $salt . sha1($salt . $plainText);
    return $salt . sha1( str: $salt . $plainText);

}
```

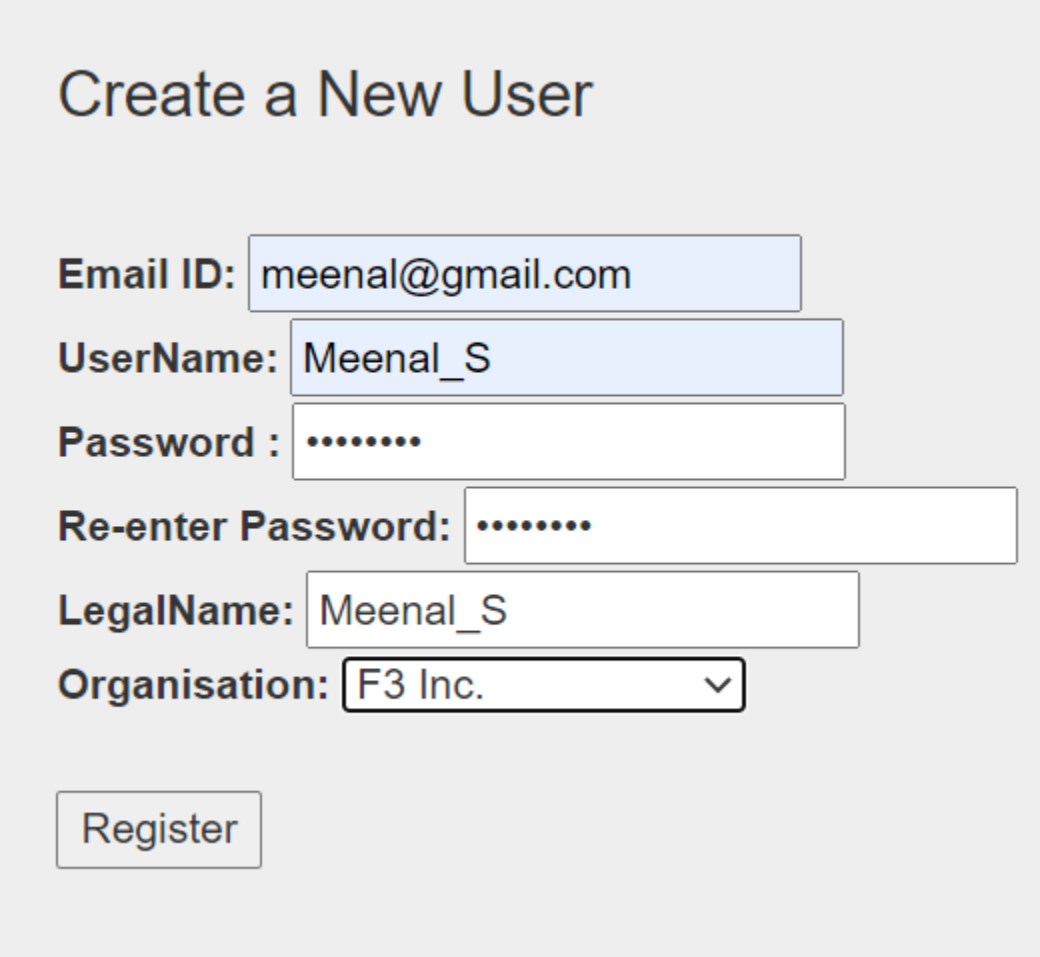
3. Create a Registration page for new users.



4. Create a Sign-in function.

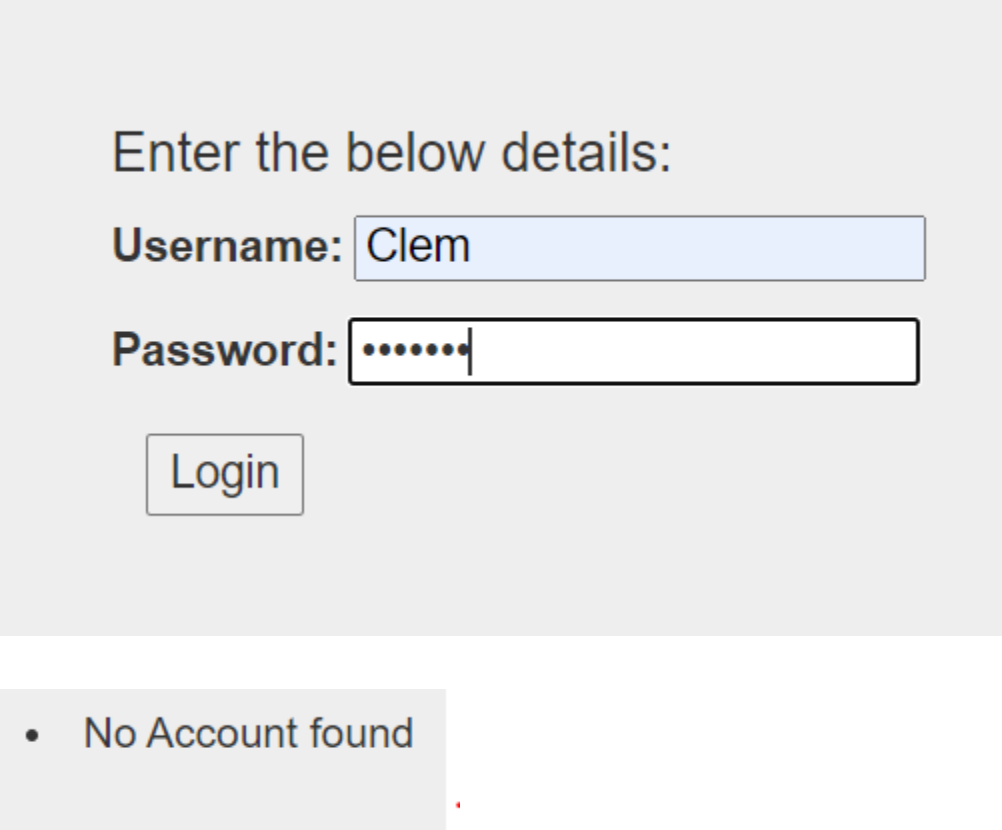


5. Username and email must be unique for signup.



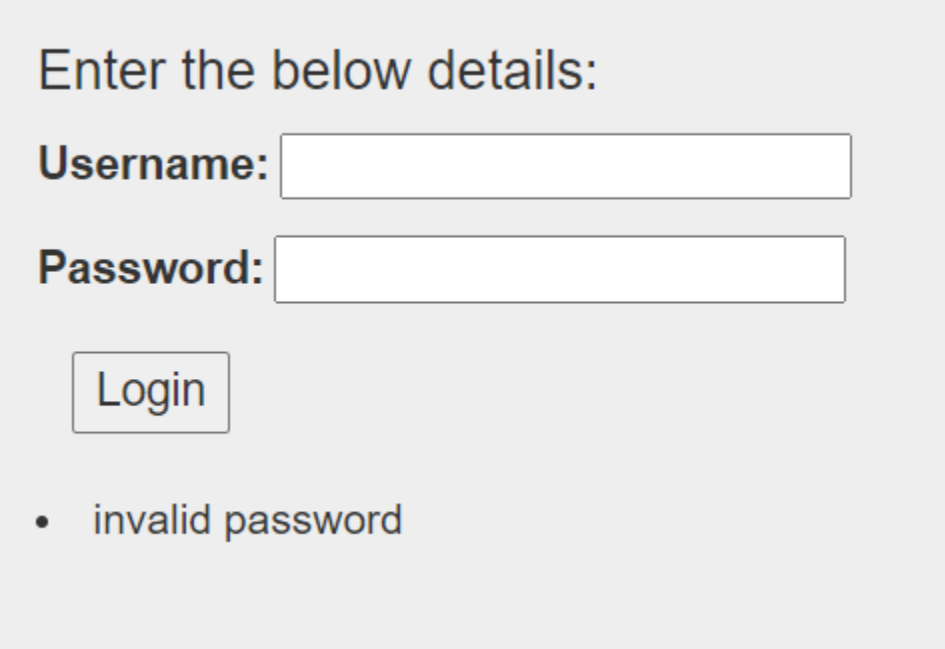
- Username/Email Exist

6. Error Message if Username does not exist.

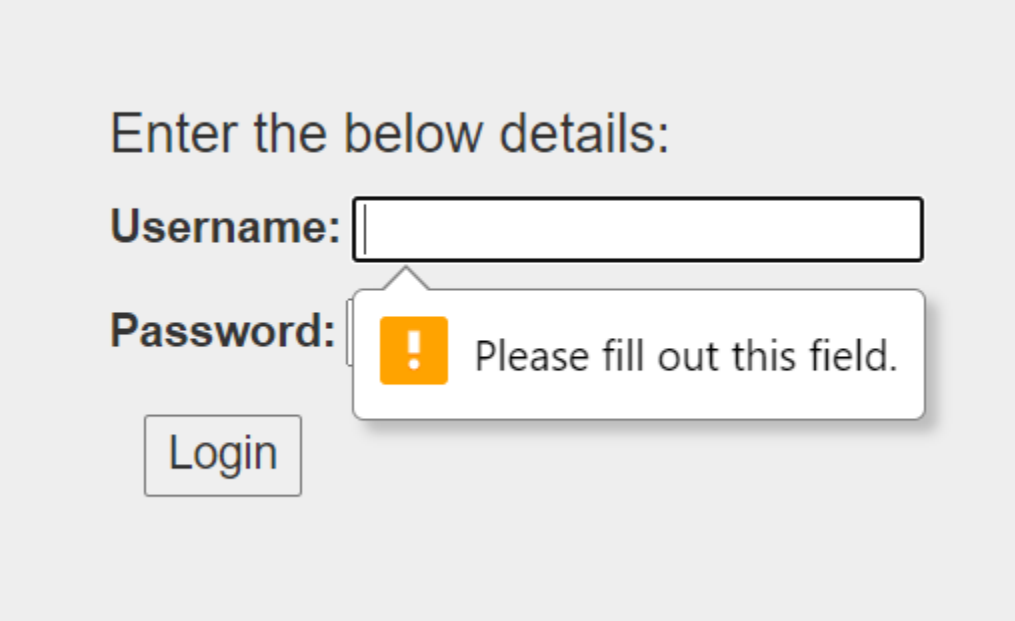


user_id	email	username
4	cmgtuv@mymailcr.com	chrismg
5	upo1342@fw025.com	ulripo
6	ultis2@fw025.com	ultis
7	tymou135@fw025.com	tyrone3
8	cand7ysm@fw025.com	candiceM
9	upo1342@mymailcr.com	upo1342
10	ekpe70@fw025.com	ericpeters
11	grancc@mymailcr.com	grant
12	wrets55@fw025.com	wilrets
13	ytumb00@fw025.com	yymb
14	stryesm@mymailcr.com	stryerM
32	meenal@gmail.com	Meenal

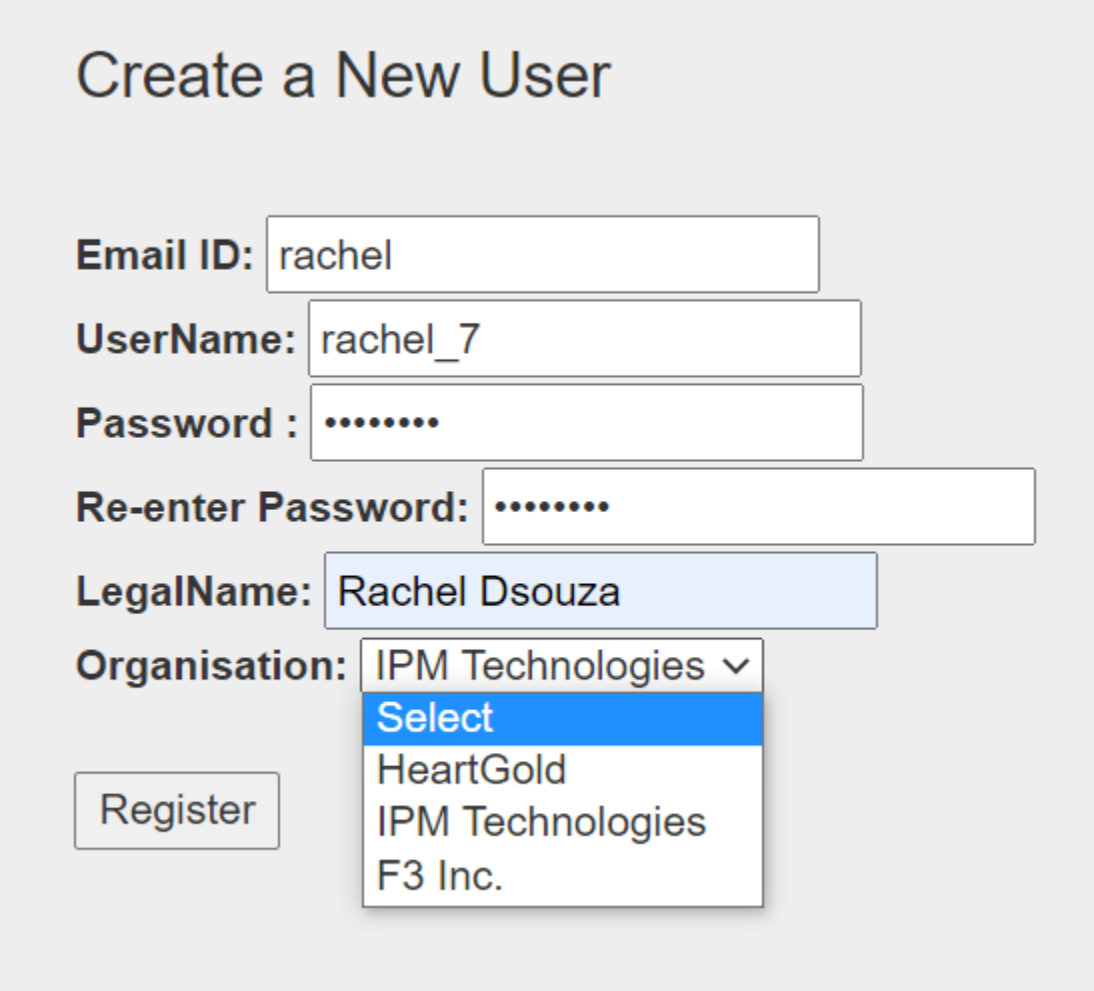
7. Error message for incorrect password



8. Mandatory Fields



9. Email Validation and dropdown to select existing organizations



- Invalid email format

10. View user's submissions:

User can view their submissions based on populated data in submission.csv.

My Submission Records:

SID	Path	Class Name	UID
2	20_newsgroups/talk.politics.mideast/76248	talk.politics.mideast	1
30	20_newsgroups/talk.politics.mideast/76428	talk.politics.mideast	1
35	20_newsgroups/talk.politics.mideast/76073	talk.politics.mideast	1
43	20_newsgroups/talk.politics.mideast/76488	talk.politics.mideast	1
49	20_newsgroups/talk.politics.mideast/76486	talk.politics.mideast	1
53	20_newsgroups/talk.politics.mideast/76029	talk.politics.mideast	1
59	20_newsgroups/talk.politics.mideast/77359	talk.politics.mideast	1
72	20_newsgroups/talk.politics.mideast/76075	talk.politics.mideast	1
73	20_newsgroups/talk.politics.mideast/76411	talk.politics.mideast	1
77	20_newsgroups/talk.politics.mideast/76418	talk.politics.mideast	1

11. View Organisation submissions.

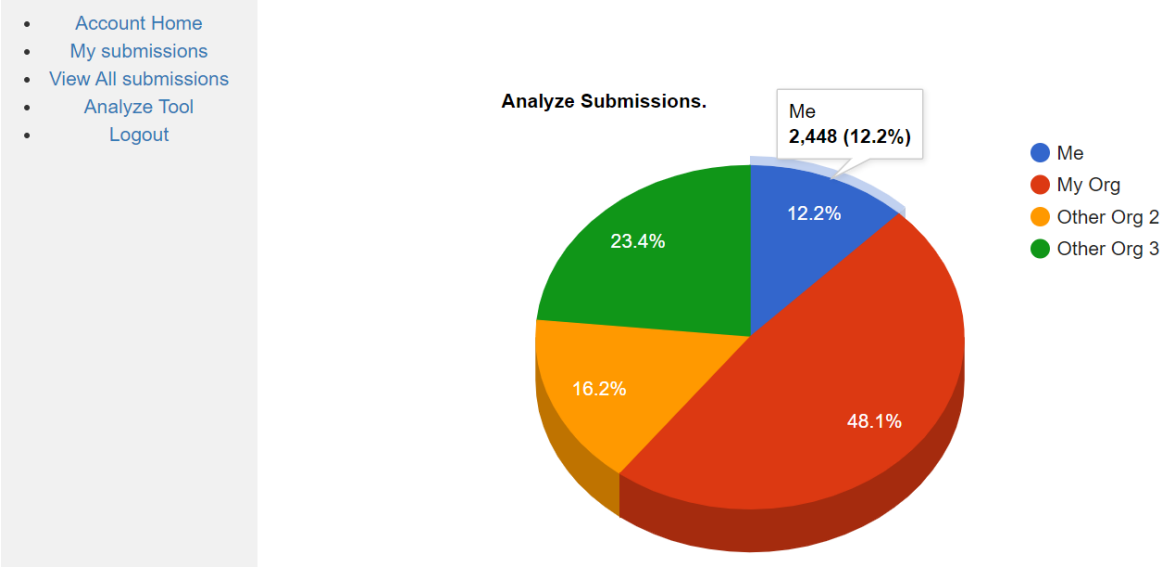
User can view their submissions for their organisation.

View All Submission Records:

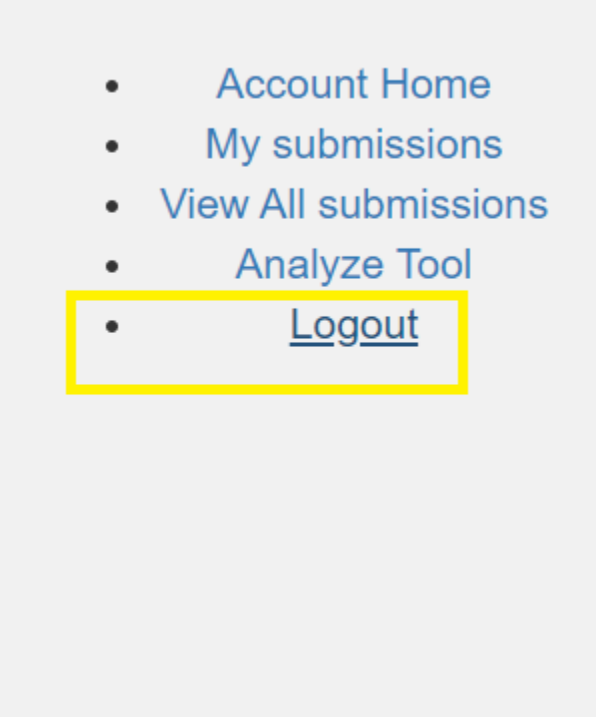
SID	Path	Class Name	UID	Organis
2	20_newsgroups/talk.politics.mideast/76248	talk.politics.mideast	1	1
30	20_newsgroups/talk.politics.mideast/76428	talk.politics.mideast	1	1
35	20_newsgroups/talk.politics.mideast/76073	talk.politics.mideast	1	1
43	20_newsgroups/talk.politics.mideast/76488	talk.politics.mideast	1	1
49	20_newsgroups/talk.politics.mideast/76486	talk.politics.mideast	1	1
53	20_newsgroups/talk.politics.mideast/76029	talk.politics.mideast	1	1
59	20_newsgroups/talk.politics.mideast/77359	talk.politics.mideast	1	1
72	20_newsgroups/talk.politics.mideast/76075	talk.politics.mideast	1	1
73	20_newsgroups/talk.politics.mideast/76411	talk.politics.mideast	1	1
77	20_newsgroups/talk.politics.mideast/76418	talk.politics.mideast	1	1

12. Analyze Tool

User can compare their data with the global population of submissions. (using Google Chart)

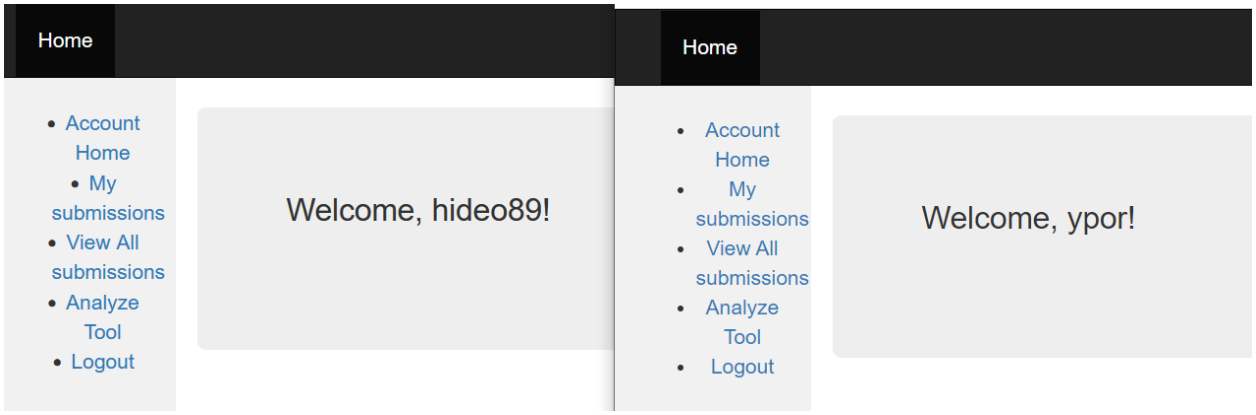


13. Create logout function.



14 Supports Parallel entries.

Multiple users can log in at the same time.



Installation Guide

Installation steps in Docker:

1. Download project source code from Github.Link :

   GitHub Main link: https://github.com/meenalsawant017/Frontend_Project

   PHP Source Code: https://github.com/meenalsawant017/Frontend_Project/tree/main/SourceCode/Email_Classification_System

   SQL Database: https://github.com/meenalsawant017/Frontend_Project/blob/main/SourceCode/email_classification.sql

2. Pull docker image.

   ```
   docker pull tomsik68/xampp
   ```

3. Run docker image.

```
docker run --name myXampp -p 41061:22 -p 41062:80 -d -v path_of_downloaded_github_code :/www tomsik68/xampp
```

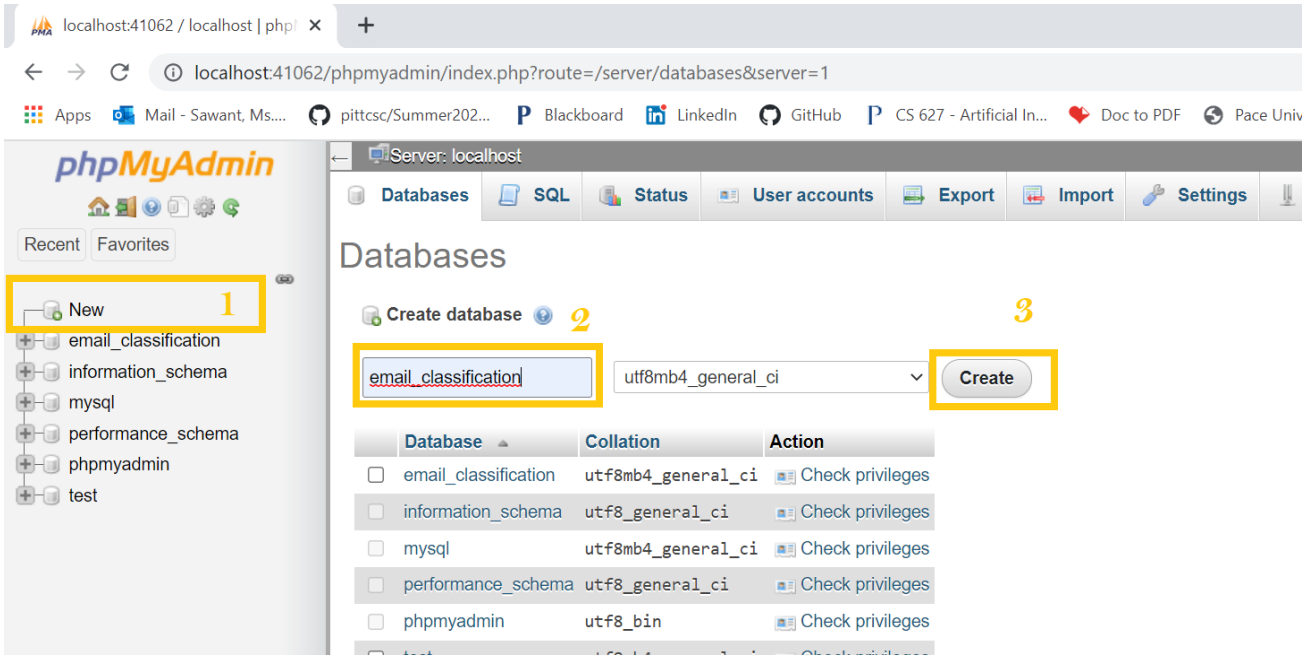
Example: `docker run --name myXampp -p 41061:22 -p 41062:80 -d -v C:\Users\XXX\Desktop\Frontend_Project-main\Frontend_Project-main\SourceCode\Email_Classification_System:/www tomsik68/xampp`

4. Start PHP admin

http://localhost:41062/phpmyadmin/

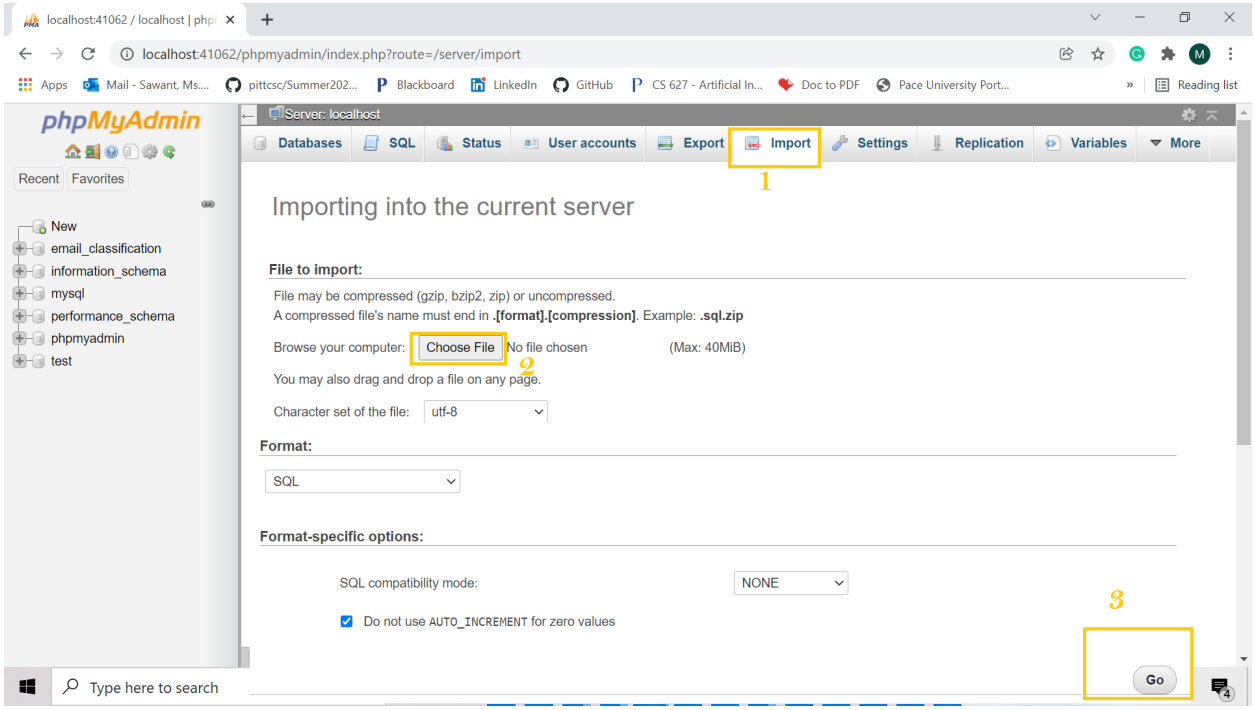
5. Create a Database.

Create the database with the name "***email_classification***". (Follow steps 1, 2, 3 as shown below).



6. Import the data.

Import the data using file
"https://github.com/meenalsawant017/Frontend_Project/blob/main/SourceCode/email_classification.sql"
(downloaded from GitHub in step 1). Follow the Import steps shown below (1,2,3)
Import Tab ->ChooseFile ->email_classification.sql->Go



7. Open application in the browser

http://localhost:41062/www/index.php