



Lab 2

Login and Registration form

Objectives:

- Dealing with PhpMyAdmin and learning basic SQL commands.
- Integrating Web applications with database.
- Creating basic registration and login form using HTML and PHP.
- Validating forms using JavaScript.

phpMyAdmin:

PhpMyAdmin is a **free software tool written in PHP to handle the administration of MySQL Database with the use of a web browser.**

It supports wide range of MySQL operations

First: Run PhpMyAdmin:

- Open XAMPP start the Apache and MySQL
- Open the browser and type into the url: **http://localhost/phpmyadmin**
- By default browser search for service using port 80, if you are using different port for apache (ex: 8080)
type : **http://localhost:8080/phpmyadmin** instead

Second: Create Database:

1. Create Your Database using **DDL statements** : **data definition language**

Open the "SQL" tab in phpmyadmin and type this DDL SQL query to create database:

```
Create database registration;
```



2. Create Your tables :

User (user_id, email, name, password, registration_date)

Note: make sure that field **user_id** is auto increment to let the system automatically generate them and will be incremented by 1 after each insert.

Use the following DDLs to create table:

Use registration;

```
create table user(  
user_id int not null auto_increment,  
email varchar(225) not null,  
name varchar(225) not null,  
password varchar(225) not null,  
registration_date timestamp default current_timestamp,  
  
PRIMARY KEY (user_id)  
);
```

3. To check your table is created correctly

On the left sidebar press on reload and click on your database, it should expand and your table should appear.

Then use the following DML insert statement to insert a new record into user table from your phpMyAdmin page :

```
INSERT INTO user (email, name ,password) VALUES  
( 'test@test.com', 'test', 'testpassword' );
```

DML: data manipulation language



Third: Create The Web App :

1. Create a web page that allows registration and login.
2. Login will ask the user to insert his **email** and **password**.

Make client side validations using **JavaScript**.

- Email and password are not empty.

PHP code should connect to MySQL database and run a query to select user data by the provided email and password.

Hint: <http://www.phponwebsites.com/2014/07/php-mysql-login-validation.html>

3. Registration will ask the user to insert name, email, password and confirm password.

Make client side validations using **JavaScript**

- Username, email, password and confirm password are not empty, display which field is empty if any.
- Email must be unique (Two users can never have same email), display "Email Already Exists" on registering with an existing email.
- Password and confirm password fields must be the same.

PHP code should connect to MySQL database and run a query to insert a new record in the **User** table.

4. After submission of both forms. If all user inputs are valid, it should display **Welcome {Name}**. If any input is invalid or empty redirect to the same page and a message should be displayed describing the errors.



Bonus:

- Email has a valid format
- Saving encrypted MD5 password.
- Using CSS to format the forms.
- Submitting the form using Ajax.

Policies :

- You should work individually.
- If 2 or more copies are discovered, all copies will lose all the marks of year work.
Hence, it is better to deliver nothing than delivering a copy.
- Late submission is allowed only for one week with 80% of the total mark. No late submission is allowed after that.

