

# **Introduction**

## **1.1 Project Overview**

A web-based application to store innovation information called “Asian Innovation Council” will be developed for internet user. The focus is to build a platform where all innovator can store their innovation in one massive library. It is expected to be a main web-based application for people around Malaysia or student in UiTM at least to access information about innovation.

The web-based application will be developed on appropriate domain. A short, memorable domain name can make the difference between creating a successful Web presence and getting lost in cyberspace. The user and innovation information from member’s registration in database will be administer by using MySQL table. The website also will integrate user and innovation information.

The web-based application will be developed by a team of two people. It required many types of programming language and tools that need to be used in order develop the website such as HTML5, CSS, PHP and Adobe Dreamweaver. The methodology uses are spiral model as to complete the website. The cost for the project is estimated about RM80.00 use to buy domain and server.

## **1.2 Problem Statement**

Many innovation show had held nowadays such as PIIC and AIS with a great amount of participant showing various kind of innovation. All the innovation are amazing and great but there are no proper place available for the participant or any innovator to share their innovation information into one platform that can be shared to the people around the globe. There are many kind of innovation are roaming in all around the cyberspace or social media that not focusing in one library that making people having hard time to search for it. This for sure will become a waste of knowledge if nothing can be done to store all the innovation into one platform so it will ease people to access it. To solve this problem, a web-based application will be developed to store all the innovation information.

### 1.3 Objective

This project will fulfil all this following objective:

1. To develop a website on appropriate domain.
2. To administer user and innovation information from member's registration.
3. To integrate user and innovation information at the developed website.

### 1.4 Scope of work

There are many types of programming language and tools that need to be used in order develop a web-based application:

- 1) HTML5 : HTML defines the structure and layout of a Web document by using a variety of tags and attributes.
- 2) CSS : CSS describes how HTML elements are to be displayed on screen, paper, or in other media.
- 3) PHP : Server-side, HTML embedded scripting language used to create dynamic Web pages.
- 4) MySQL : An open-source relational database management system (RDBMS).
- 5) XAMPP : XAMPP stands for Cross-Platform (X), Apache (A), MariaDB(M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes.
- 6) Adobe Dreamweaver : Enables the HTML programmer to build complex websites using HTML, JavaScript and server-side programming languages.

## Project Progress and Result

### 2.1 User Interface / GUI

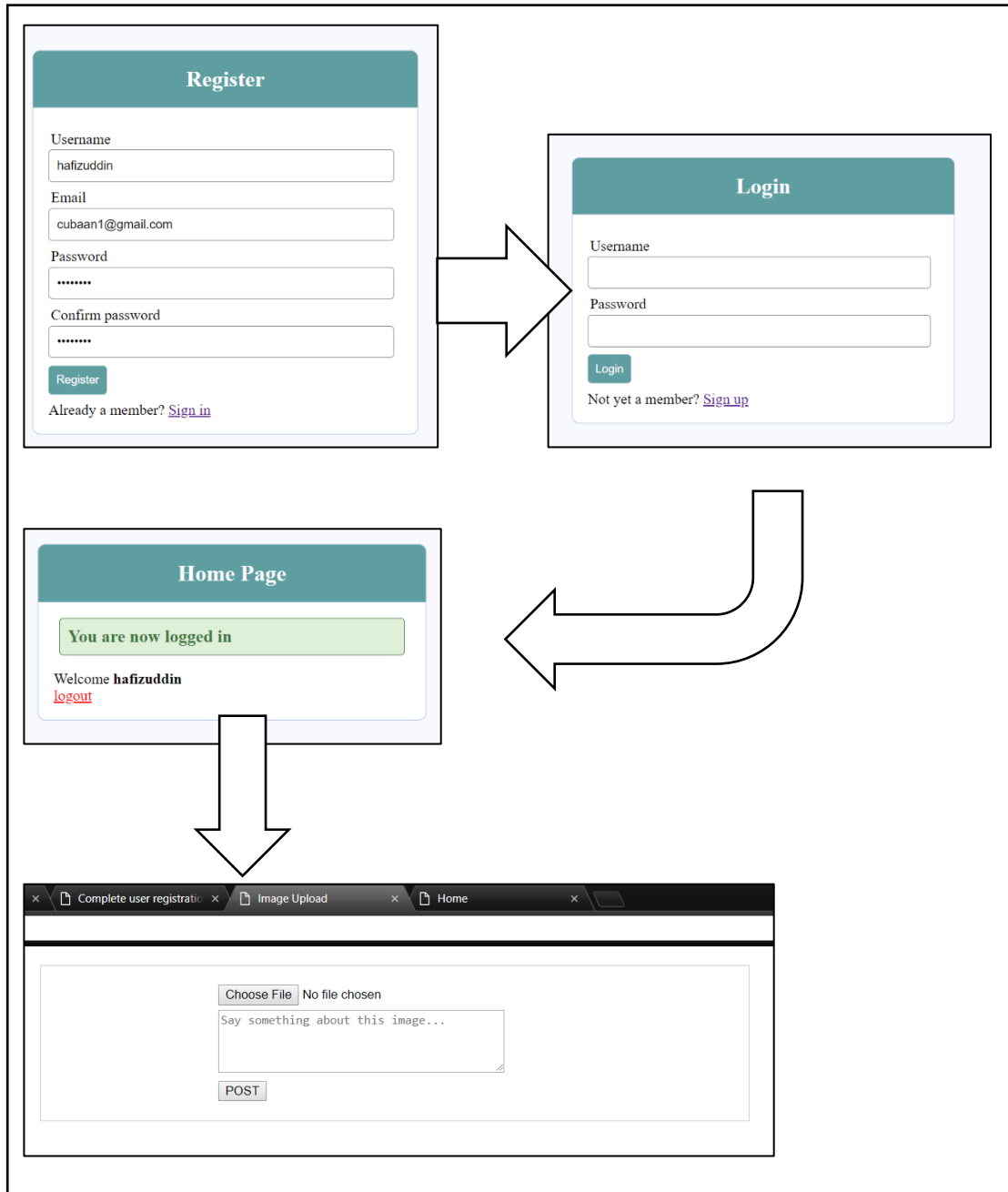


Figure 2.01 User Interface

## 2. 2 Block Diagram

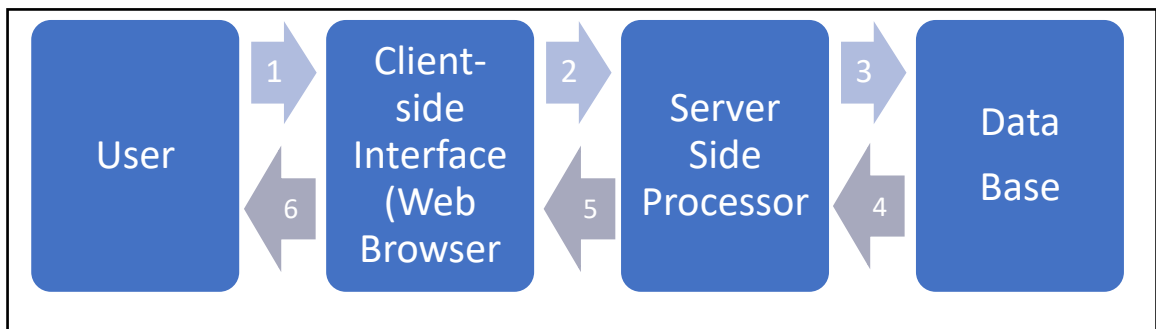
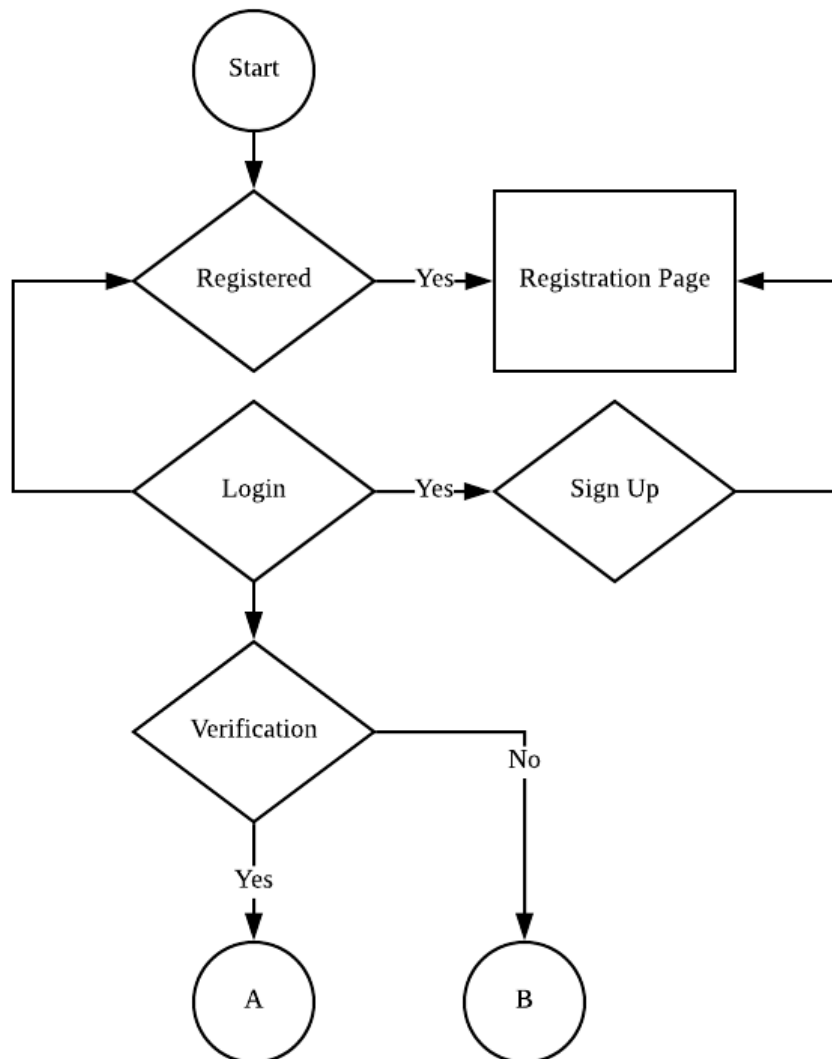
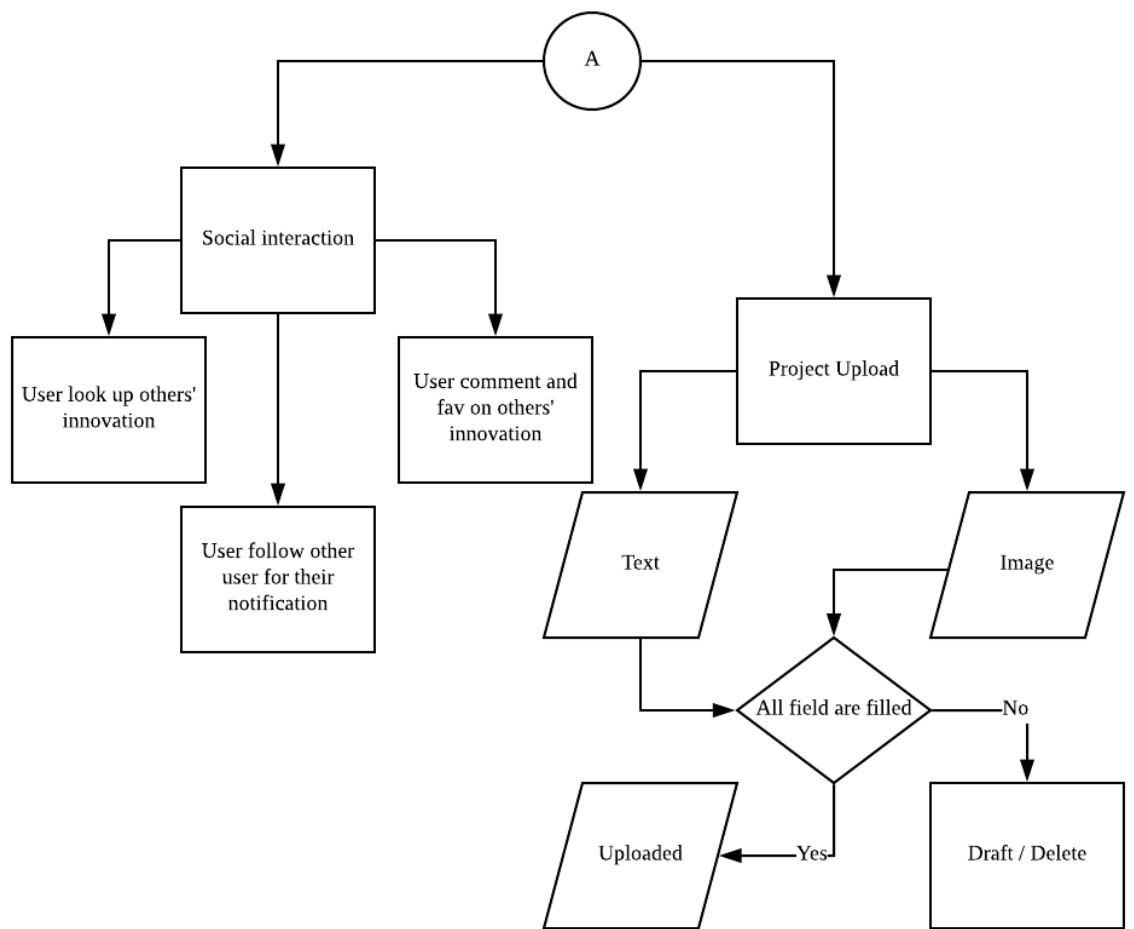


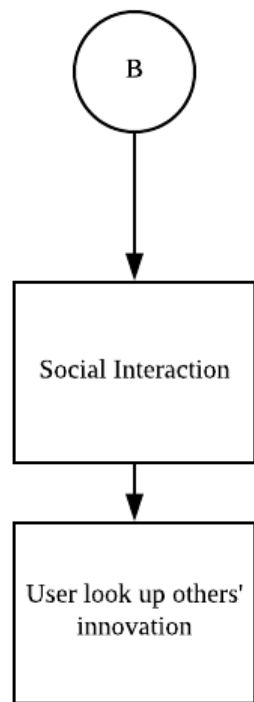
Figure 1.02 Block Diagram

1. User defines a Query using the Query interface.
2. Query interface sends the query to a server-side processing agent.
3. Server-side agent respond to the query using some data source or other backend service.
4. Data source return Query result.
5. Server-side agent return Query result.
6. Query result are displayed to the user.

## 2.3 System Operation









## 2.4 Preliminary Result

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop Primary Unique More
2	username	varchar(100)	latin1_swedish_ci		No	None			Change Drop Primary Unique More
3	email	varchar(100)	latin1_swedish_ci		No	None			Change Drop Primary Unique More
4	password	varchar(100)	latin1_swedish_ci		No	None			Change Drop Primary Unique More

Figure 2.02 MySQL table

First we create database for the member database as registration using Mysql thru PHPMyadmin, we add a table called users and add id, username, email and password in the table.

```
<?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>
```

Figure 2.03 register.php codes

First we do register.php, this is where user will register their membership. The code that received the form of data is written in the server.php file, it put at the top of the file. We also put errors.php files to display error.

```

* {
  margin: 0px;
  padding: 0px;
}
body {
  font-size: 120%;
  background: #F8F8FF;
}

.header {
  width: 30%;
  margin: 50px auto 0px;
  color: white;
  background: #5F9EA0;
  text-align: center;
  border: 1px solid #80C4DE;
  border-bottom: none;
  border-radius: 10px 10px 0px 0px;
  padding: 20px;
}

form, .content {
  width: 30%;
  margin: 0px auto;
  padding: 20px;
  border: 1px solid #80C4DE;
  background: white;
  border-radius: 0px 0px 10px 10px;
}

.input-group {
  margin: 10px 0px 10px 0px;
}

.input-group label {
  display: block;
  text-align: left;
  margin: 3px;

```

Figure 2.04 style.css codes 1

```

}

.input-group input {
  height: 30px;
  width: 93%;
  padding: 5px 10px;
  font-size: 16px;
  border-radius: 5px;
  border: 1px solid gray;
}

.btn {
  padding: 10px;
  font-size: 15px;
  color: white;
  background: #5F9EA0;
  border: none;
  border-radius: 5px;
}

.error {
  width: 92%;
  margin: 0px auto;
  padding: 10px;
  border: 1px solid #a94442;
  color: #a94442;
  background: #f2dede;
  border-radius: 5px;
  text-align: left;
}

.success {
  color: #3c763d;
  background: #dfff0d8;
  border: 1px solid #3c763d;
  margin-bottom: 20px;
}

```

Figure 2.05 style.css codes 2

This is the style.css file, it work to improve the web to look user friendly.

```

<?php
session_start();

// initializing variables
$username = "";
$email    = "";
$errors = array();

// connect to the database
$db = mysqli_connect('localhost', 'root', '', 'registration');

// REGISTER USER
if (isset($_POST['reg_user'])) {
    // receive all input values from the form
    $username = mysqli_real_escape_string($db, $_POST['username']);
    $email = mysqli_real_escape_string($db, $_POST['email']);
    $password_1 = mysqli_real_escape_string($db, $_POST['password_1']);
    $password_2 = mysqli_real_escape_string($db, $_POST['password_2']);

    // form validation: ensure that the form is correctly filled ...
    // by adding (array_push()) corresponding error unto $errors array
    if (empty($username)) { array_push($errors, "Username is required"); }
    if (empty($email)) { array_push($errors, "Email is required"); }
    if (empty($password_1)) { array_push($errors, "Password is required"); }
    if ($password_1 != $password_2) {
        array_push($errors, "The two passwords do not match");
    }

    // first check the database to make sure
    // a user does not already exist with the same username and/or email
    $user_check_query = "SELECT * FROM users WHERE username='$username' OR email='$email'";
    $result = mysqli_query($db, $user_check_query);
    $user = mysqli_fetch_assoc($result);

    if ($user) { // if user exists
        if ($user['username'] == $username) {
            array_push($errors, "Username already exists");
        }
    }
}

```

Figure 2.07 server.php codes 1

```

}

// Finally, register user if there are no errors in the form
if (count($errors) == 0) {
    $password = md5($password_1);//encrypt the password before saving in the database

    $query = "INSERT INTO users (username, email, password)
VALUES('$username', '$email', '$password')";
    mysqli_query($db, $query);
    $_SESSION['username'] = $username;
    $_SESSION['success'] = "You are now logged in";
    header('location: index.php');
}
}

// ...

```

Figure 2.08 server.php codes 2

```

// LOGIN USER
if (isset($_POST['login_user'])) {
    $username = mysqli_real_escape_string($db, $_POST['username']);
    $password = mysqli_real_escape_string($db, $_POST['password']);

    if (empty($username)) {
        array_push($errors, "Username is required");
    }
    if (empty($password)) {
        array_push($errors, "Password is required");
    }

    if (count($errors) == 0) {
        $password = md5($password);
        $query = "SELECT * FROM users WHERE username='$username' AND password='$password'";
        $results = mysqli_query($db, $query);
        if (mysqli_num_rows($results) == 1) {
            $_SESSION['username'] = $username;
            $_SESSION['success'] = "You are now logged in";
            header('location: index.php');
        } else {
            array_push($errors, "Wrong username/password combination");
        }
    }
}
}
}

```

Figure 2.09 server.php codes 3

This is the server.php file that will do the job to insert data form by user into the database.

```

<?php if (count($errors) > 0) : ?>
    <div class="error">
        <?php foreach ($errors as $error) : ?>
            <p><?php echo $error ?></p>
        <?php endforeach ?>
    </div>
<?php endif ?>

```

Figure 2.10 errors.php codes

This errors.php file that will display the error that the user if they make any mistake while registering or log in

```

<?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>Login</h2>
    </div>

    <form method="post" action="login.php">
        <?php include('errors.php'); ?>
        <div class="input-group">
            <label>Username</label>
            <input type="text" name="username" />
        </div>
        <div class="input-group">
            <label>Password</label>
            <input type="password" name="password" />
        </div>
        <div class="input-group">
            <button type="submit" class="btn" name="login_user">Login</button>
        </div>
        <p>
            Not yet a member? <a href="register.php">Sign up</a>
        </p>
    </form>
</body>
</html>

```

Figure 2.11 login.php codes

This is the login.php files, it provides a system that only for registered member where their data had been saved in the database.

```

<?php
    session_start();

    if (!isset($_SESSION['username'])) {
        $_SESSION['msg'] = "You must log in first";
        header('location: login.php');
    }
    if (isset($_GET['logout'])) {
        session_destroy();
        unset($_SESSION['username']);
        header("location: login.php");
    }
?>
<!DOCTYPE html>
<html>
<head>
    <title>Home</title>
    <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>

<div class="header">
    <h2>Home Page</h2>
</div>
<div class="content">
    <!-- notification message -->
    <?php if (isset($_SESSION['success'])) : ?>
        <div class="error success" >
            <h3>
                <?php
                    echo $_SESSION['success'];
                    unset($_SESSION['success']);
                ?>
            </h3>
        </div>
    <?php endif ?>

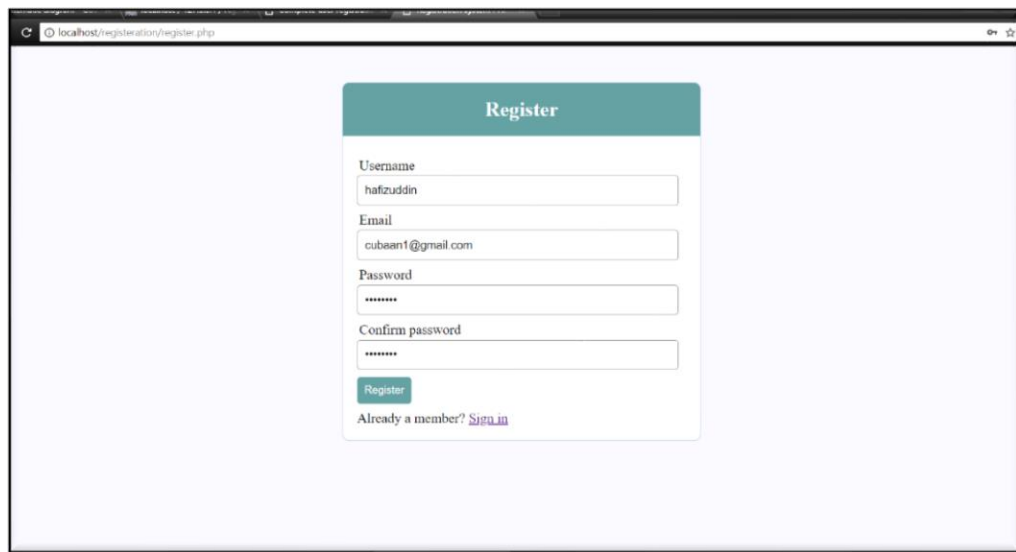
    <!-- logged in user information -->
    <?php if (isset($_SESSION['username'])) : ?>
        <p>Welcome <strong><?php echo $_SESSION['username']; ?></strong></p>
        <p> <a href="index.php?logout='1'" style="color: red;">logout</a> </p>
    <?php endif ?>
</div>

```

Figure 2.12 index.php codes

This is the index.php files, it will be the first to show up in the web when the user open the website.

## 2.5 Simulation



The screenshot shows a web browser window with the address bar displaying 'localhost/registeration/register.php'. The main content is a registration form with a teal header 'Register'. The form contains four input fields: 'Username' with the value 'hafizuddin', 'Email' with the value 'cubaan1@gmail.com', 'Password' with masked characters '\*\*\*\*\*', and 'Confirm password' also with masked characters '\*\*\*\*\*'. Below the fields is a teal 'Register' button and a link that says 'Already a member? [Sign in](#)'.

Figure 2.13 Register page

First we going to register username, email and password in the localhost/registeration/register.php. If succeed then all the data that we submitted form the form will be saved in the database at mysql.

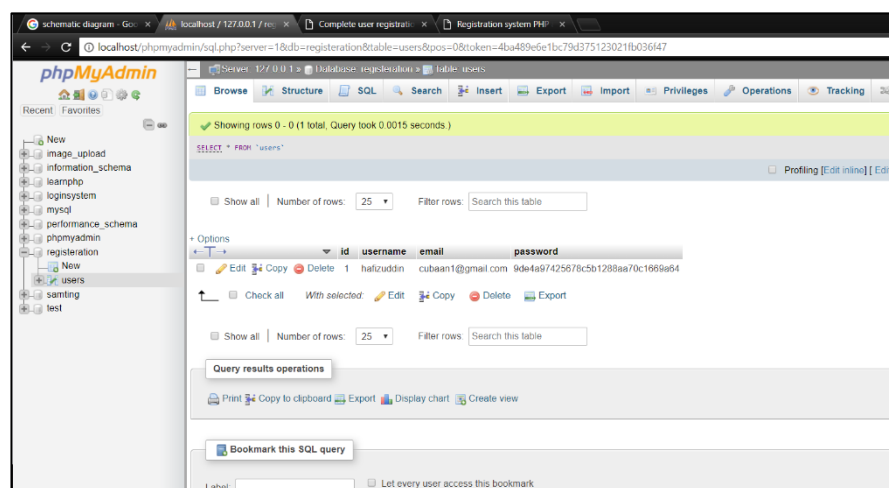


Figure 2.14 MySQL client phpMyAdmin

Here all the submitted data been stored in mysql table. Noticed that password had been hashed for security of the member.

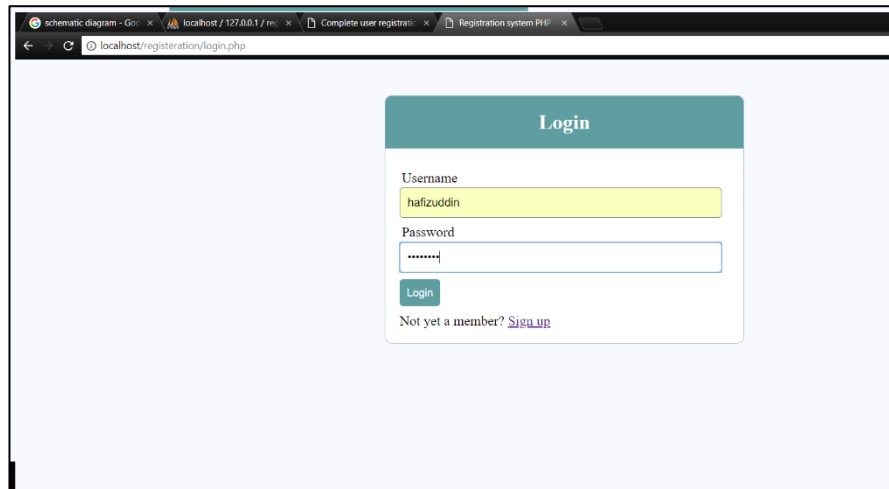


Figure 2.15 Log in page

To approve that that the data stored can be retrieved, we try to log in using the same username and password in database.

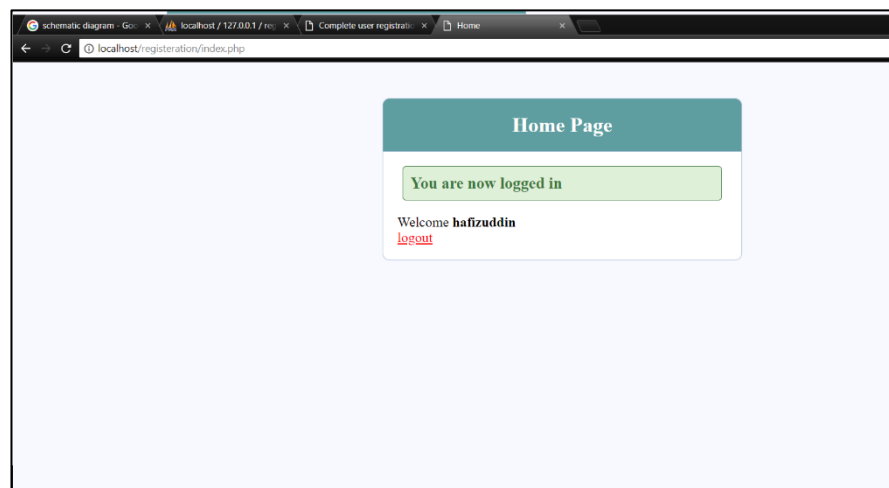


Figure 2.16 Success log in

The site show that our logged in successful so our simulation for member database are success.

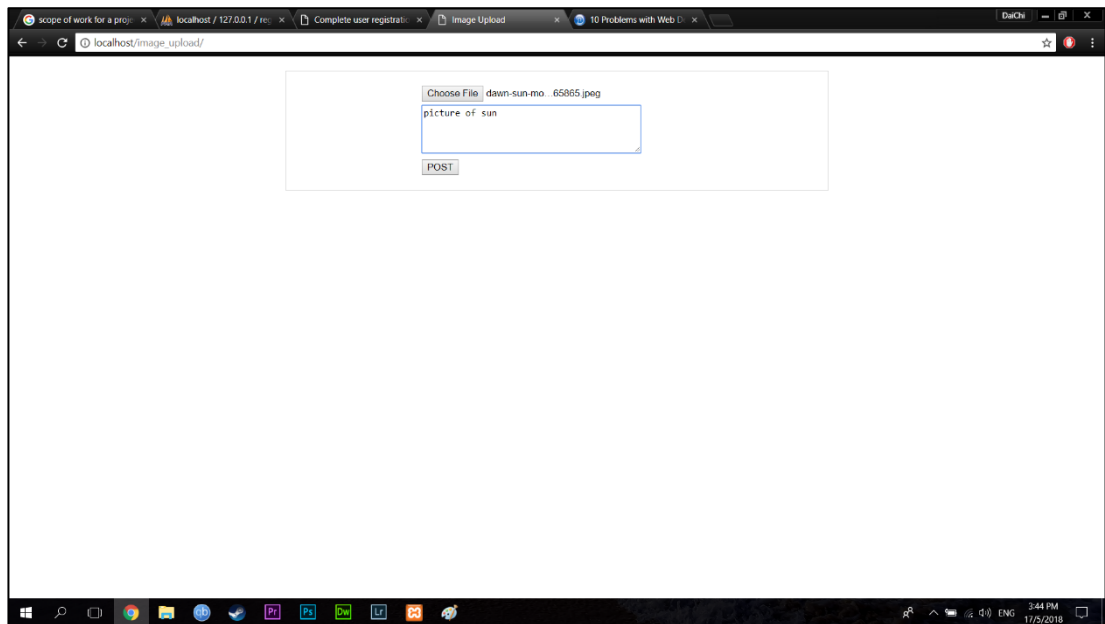


Figure 2.17 Upload image

As for the innovation database, it will be storing data such as picture and text so we. In the picture we upload a picture and some text that will be store in database.

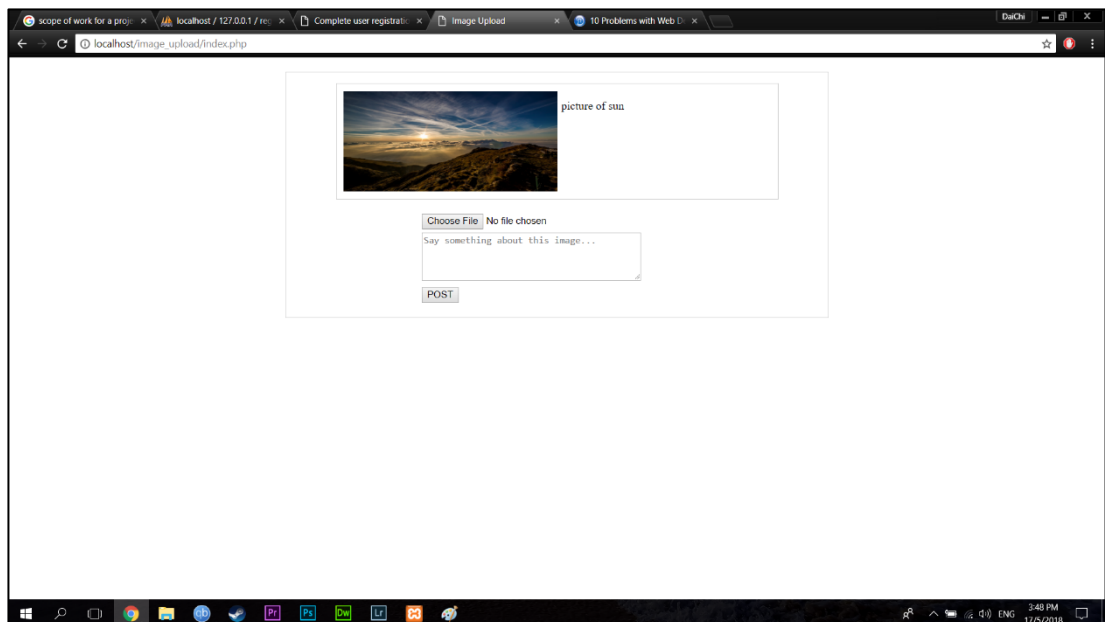


Figure 2.18 Picture shown

To prove that the picture stored in the database, we retrieve it in the website to show that it was success.



## **2.6 Problem Encountered and the Solutions**

Our main problem is we were difficult to find proper time to a set meet up with our supervisor. The date of a submission also clashed with test and classes. So it is hard to get our progress going smoothly. Then, we discussed and compared each other timetable to make our work get along. We divided tasks and work that need to be done in a short time. Because if we have a proper time before we hands in, we also can have some time to meet.

Secondly, as this is our first time of doing an engineering project, so we lack of knowledge in order to improve our web-application. There were the dos and don'ts in anything, same goes to our project. We need to do some improvise to make it better in quality. This problem we faced lead us to a solution where we find our supervisor of course, and other lecturer. We also studied by our own at library using specified books.

Last but not least, we have faced a lot of problems in our simulation. At first it may seem easy, but as soon as we started there were many parts of coding that were left blank. As soon as we figured out its full coding, turns out the coding are a bit messy. Other than that, web-application with its function require more than client-server coding. So for these things, first we use a software called Adobe Dreamweaver to make our coding systematic. Next we also use MySQL client PhpMyAdmin, XAMPP, as a medium to connect to our local host.

## Project Plan

Table 3.1 Project Plan

	Duration (Week)															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
FYP 2 Briefing	1	/														
Meeting up with supervisor	14		/	/	/	/	/	/	/	/	/	/	/	/	/	/
Team meet-up	14		/	/	/	/	/	/	/	/	/	/	/	/	/	/
Research and buying server cloud and domain	3									/	/	/				
Perfecting the coding	14		/	/	/	/	/	/	/	/	/	/	/	/	/	/
Research on troubleshooting and problem solving	6								/	/	/	/	/	/		
<b>First Draft (Chapter 1 and 2)</b>	1					/										
<b>Second Draft (Chapter 3)</b>	1								/							
<b>Final Draft</b>	1											/				
<b>Presentation</b>	1														/	
<b>Report Submission</b>	1															/

## **CONCLUSION**

As for the conclusion, develop a website on a proper domain does make the difference between creating a successful Web presence and getting lost in cyberspace. Then, administer user and innovation information from member's registration using MySQL table really help making the data organised and easy to trace. The website are also able to integrate user and innovation information.

## REFERENCES

- [1] Robert Mening, 'How to Make a Website', [ONLINE] <https://websitesetup.org> [Accessed: 13- March- 2018]
- [2] Chris David Milss, 'Publishing Your Website', [ONLINE] <https://developer.mozilla.org> [Accessed: 16- March- 2018]
- [3] Unknown, 'Quick Tag Reference', [ONLINE] <https://www.littlewebhut.com> [Accessed: 10- March- 2018]
- [4] Michael Alex, Journal of Visual Impairment & Blindness, Volume 99, Issues 7 – 12, 22 Jul 2009
- [5] Vikram Vaswani, PHP: A Beginner's Guide, The McGraw Company, Aug 2009