Mini Project - 2

|  |  |
| --- | --- |
| **Student Name/ID Number:** | Francis Roel L. Abarca – BDSE-0922-113 |
| **Unit Number and Title:** | ACWD Module 4 – Database Design & Implementation |
| **Academic Year:** |  |
| **Unit Assessor:** |  |
| **Project Title:** | Implementing a Database for Community Portal |
| **Issue Date:** |  |
| **Submission Date:** | 1/19/2023 |
| **Internal Verifier Name:** |  |
| **Date:** | 1/19/2023 |

|  |
| --- |
| **Learner declaration** |
| I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.  Student signature: Date: |

|  |
| --- |
| **Purpose of this project** |
| **Purpose of this project**  To demonstrate your capabilities in the following areas:   * Ability to Implement a MySQL Database based on design created in Mini Project 1 |
| **Submission Format** |
| 1. Screen capture of Database Table structure in PhpMyAdmin 2. Queries used by Web pages in Tabular Format |
| **Project Brief & Guidance** |
| **Scenario:**  **Refer to the Project Scenario for the Module Project**  You have been approached by ‘ABC Jobs Pte Ltd’ as a website developer to develop a community portal for Software Developers. The project will be carried over through Module 3, Module 4, Module 5 and Capstone project. For this module the scope is to Design, Develop, Implement & Document Struts Framework Website.  The Scope of the Project is to design a Community Portal Similar to Linkedin.com. Users will be able to register in the portal using the Registration Page. Users of the portal can search for other users using various parameters such as First Name, Last Name, Company Name, City & Country. Users will be able to view the Public Profile of users after searching them. The portal allow users to login, request for forgotten password and Update their profile information  The scope of the mini project is to implement the database in MySQL.  **The overview of the project is as below**  There are 2 types of users in this Community portal. They are   1. Software Programmer 2. Administrator   **Software Programmer should be able to perform following functions in the portal**   1. Allow the programmers to register in the portal, show a thank you page & send a registration confirmation email. 2. Search & Find Other Programmers after login and view their profile. 3. Provide Login Page 4. Provide password retrieval functionality. |

|  |
| --- |
| 1. Update their Profile after logging in.   Following Functionality is part of Database Design & Will be enhanced as features as required in the Capstone Project, Depending on time availability.   1. Send Messages to Each Other on the Portal 2. Create Threads & Post Replies to a Thread 3. Post Job Opportunities in the Portal   **Administrator should be able to perform following functions in the portal**   1. Administer user data. 2. Send bulk email inviting programmers to register on the community portal   **The portal consist of the following Key pages (For Reference)**   1. Community Portal Home Page 2. Registration Page 3. Registration Confirmation Page 4. Update Profile Page 5. Search Users Page 6. List Search Results 7. Public Profile Page 8. Registration Confirmation Email 9. Login Page 10. Forget Password Page 11. Design the Forget Password Confirmation Page   Following Functionality is part of Database Design and will not be part of development & Will be enhanced as features are required in the Capstone Project, Depending on time availability.   1. Send Messages 2. Read Messages 3. Post in Message Board 4. List Message Board 5. Read A Thread 6. Post Job Opportunities 7. List Job Opportunities & Responses   **The scope of this assignment**  The scope is to Implement the database in MySQL   * 1. Create the tables based on design in Mini Project 1   CREATE TABLE Admin  (  `#admin\_id` INT NOT NULL,  `#user\_id` INT NOT NULL,  admin\_password VARCHAR(15) NOT NULL,  admin\_email VARCHAR(20) NOT NULL,  PRIMARY KEY (`#admin\_id`)  );  CREATE TABLE Bulk\_Mail  (  `#email\_id` INT NOT NULL,  `#admin\_id` INT NOT NULL,  send\_date VARCHAR(25) NOT NULL,  mail\_subject VARCHAR(25) NOT NULL,  mail\_content VARCHAR(200) NOT NULL,  PRIMARY KEY (`#email\_id`),  FOREIGN KEY (`#admin\_id`) REFERENCES Admin(`#admin\_id`)  );  CREATE TABLE User  (  `#user\_id` INT NOT NULL,  user\_name VARCHAR(50) NOT NULL,  user\_email VARCHAR(20) NOT NULL,  user\_role VARCHAR(20) NOT NULL,  user\_password VARCHAR(15) NOT NULL,  user\_country VARCHAR(20) NOT NULL,  `#admin\_id` INT NOT NULL,  PRIMARY KEY (`#user\_id`),  FOREIGN KEY (`#admin\_id`) REFERENCES Admin(`#admin\_id`)  );  CREATE TABLE Message  (  `#msg\_id` INT NOT NULL,  `#sender\_id` INT NOT NULL,  `#receipient\_id` INT NOT NULL,  msg\_time\_date DATETIME NOT NULL,  msg\_content VARCHAR(100) NOT NULL,  PRIMARY KEY (`#msg\_id`),  FOREIGN KEY (`#sender\_id`) REFERENCES User(`#user\_id`),  FOREIGN KEY (`#receipient\_id`) REFERENCES User(`#user\_id`)  );  CREATE TABLE Jobs  (  `#job\_id` INT NOT NULL,  `#user\_id` INT NOT NULL,  job\_name VARCHAR(50) NOT NULL,  job\_companyname VARCHAR(50) NOT NULL,  job\_address VARCHAR(50) NOT NULL,  job\_salary DOUBLE) NOT NULL,  PRIMARY KEY (`#job\_id`),  FOREIGN KEY (`#user\_id`) REFERENCES User(`#user\_id`)  );  CREATE TABLE Experience  (  `#workexp\_id` INT NOT NULL,  prev\_job VARCHAR(50) NOT NULL,  company\_name VARCHAR(50) NOT NULL,  date\_start VARCHAR(25) NOT NULL,  date\_end VARCHAR(25) NOT NULL,  year\_count INT NOT NULL,  `#user\_id` INT NOT NULL,  PRIMARY KEY (`#workexp\_id`),  FOREIGN KEY (`#user\_id`) REFERENCES User(`#user\_id`)  );  CREATE TABLE Thread  (  `#thread\_id` INT NOT NULL,  `#user\_id` INT NOT NULL,  thread\_name VARCHAR(50) NOT NULL,  thread\_content VARCHAR(200) NOT NULL,  thread\_dateTime VARCHAR(25) NOT NULL,  PRIMARY KEY (`#thread\_id`),  FOREIGN KEY (`#user\_id`) REFERENCES User(`#user\_id`)  );   * 1. Take note of Field Name, Field Type & Size  1. User Table  * **#user\_id** int pk * User\_name varchar(50) * User\_email varchar(20) * User\_role varchar(20) * User\_password varchar(15) * User\_country varchar(20) * **#admin\_id** int  1. Jobs  * **#job\_id** int pk * **#user\_id** int * Job\_name varchar(50) * Job\_companyname varchar(50) * Job\_address varchar(50) * Job\_salary DOUBLE  1. Message  * **#msg\_id** int pk * **#sender\_id** int * **#receipient\_id** int * msg\_time\_date datetime * msg\_content varchar(100)  1. Thread  * **#thread\_id** int pk * **#user\_id** int * Thread\_name varchar(50) * Thread\_content varchar(200) * Thread\_dateTime varchar(25)  1. Work Experience  * **#workexp\_id** int pk * prev\_job varchar(50) * company\_name varchar(50) * date\_start varchar(25) * date\_end varchar(25) * year\_count int * **#user\_id** int  1. Admin  * **#admin\_id** int pk * #user\_id int * Admin\_password varchar(15) * Admin\_email varchar(20)  1. Bulk Mail  * **#email\_id** int pk * **#admin\_id** int * Send\_date varchar(25) * Mail\_subject varchar(25) * Mail\_content varchar(200)   1. Implement the database in MySQL and provide screen capture  1. User 2. Jobs 3. Message 4. Thread 5. Work Experience 6. Admin 7. Bulk Mail     1. Create a user name in MySQL which can be used by the application, give required access privileges. (Provide screen capture of the user with access privileges) 8. Create a new Account      1. Configure your new account        1. Test your connection. 2. Test your schemas and tables on the new account. |

e. Briefly list the web pages in the Community portal and queries which will be used by each of them.

|  |  |  |  |
| --- | --- | --- | --- |
| Page | Tables | Function | Query |
| 1. Log-in | User | Validate Credentials | SELECT user\_email, user\_password FROM `user`  WHERE user\_email = 'adamtheman@abc.org' AND user\_password = 'adamman1927'; |
| 1. Registration | User | Add Credentials | INSERT INTO user (user\_name, user\_email, user\_password, user\_country) VALUES (“Allan Kay”, [allankay69@gmail.com](mailto:allankay69@gmail.com), “ieu4-81ka-1a6d”); |
| 1. View Profile | User | View Credentials | SELECT user\_name, user\_email, user\_password FROM user  WHERE Email = “registered email” AND user\_name= “registered name”; |
| 1. List Search Results | Jobs | View Jobs available | SELECT \* FROM communityportal.jobs  WHERE (job\_name LIKE '%Software%' OR 'Software' = '')  AND (job\_address LIKE '%california%' OR 'california' = '')  AND (job\_salary >= 100000 OR 100000 = 0); |
| 1. Messages | Messages | Read and Send Messages | INSERT INTO messages (sender\_id, recipient\_id, msg\_content) VALUES ({sender\_id}), {receiver\_id}, ‘{msg\_content}’, ‘{sent\_at}’); |
| 1. Threads | Thread | Create Threads | INSERT INTO thread (`#user\_id`, thread\_name, thread\_content)  VALUES (3, 'Service', 'ServiceTesting'); |
| 1. Job Opportunities | Job Experience, Jobs | Show and recommend jobs based on experience. | SELECT j.\* FROM job j  JOIN experience e ON j.id = e.job\_id WHERE e.user\_id = {user\_id} GROUP BY j.id HAVING COUNT(e.job\_id) >= {year\_count}; |
| 1. Admin | Admin | Manage the Community Portal | SELECT \* FROM admin  WHERE admin\_email = {admin\_email} AND admin\_password = ‘{admin\_password}’; |
| 1. Bulk Mail | Bulk Mail | Send Bulk Mail to users | INSERT INTO bulk\_email (mail\_subject, mail\_content) VALUES (‘{mail\_subject}’, ‘{mail\_content}’); |