Mini Project - 3

|  |  |
| --- | --- |
| **Student Name/ID Number:** | Francis Roel L. Abarca – BDSE-0922-113 |
| **Unit Number and Title:** | ACWD Module 4 – Database Design & Implementation |
| **Academic Year:** | 2022-2023 |
| **Unit Assessor:** |  |
| **Project Title:** | Develop Sample Data, Queries or Report, Test & Document Database |
| **Issue Date:** |  |
| **Submission Date:** | 1/30/2022 |
| **Internal Verifier Name:** |  |
| **Date:** | 1/30/2022 |

|  |
| --- |
| **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:   * Develop Sample Data for Reports * Develop Queries & Report * Test Database * Document Database |
| **Submission Format** |
| 1. Screen capture of Sample Data for Report 2. Queries for generating reports from database 3. Detailed description of Test Methods 4. Documentation of Database |
| **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 this mini project is to generate reports, test and document database.  **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. 5. 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**   1. Create test data for all tables suitable for generating reports   a. Threads  INSERT INTO thread (user\_id, thread\_name, thread\_content) VALUES (1, "Networking Tips for Job Seekers", "Discuss strategies for building professional connections and networking effectively in the job search process.");  INSERT INTO thread (user\_id, thread\_name, thread\_content) VALUES (3, "Resume Writing and Optimization", "Tips and best practices for creating a strong resume that stands out to potential employers.");  INSERT INTO thread (user\_id, thread\_name, thread\_content) VALUES (5, "Interview Preparation 101", "We offer guidance on how to prepare for job interviews, including common questions, dress code, and body language.");  INSERT INTO thread (user\_id, thread\_name, thread\_content) VALUES (2, "Remote Work and Telecommuting", "Explore the growing trend of remote work and the pros and cons of telecommuting for both employees and employers.");  INSERT INTO thread (user\_id, thread\_name, thread\_content) VALUES (5, "Negotiating Salary and Benefits", "We provide advice on how to negotiate a fair salary and benefits package during the job offer process.");  INSERT INTO thread (user\_id, thread\_name, thread\_content) VALUES (2, "Career Advancement Strategies", "Lets discuss ways to advance your career, including professional development, networking, and building a personal brand.");  INSERT INTO thread (user\_id, thread\_name, thread\_content) VALUES (4, "Job Search Resources and Tools", "We provide information on various job search resources and tools, such as job boards, networking platforms, and career counseling services.");  INSERT INTO thread (user\_id, thread\_name, thread\_content) VALUES (1, "Career Change and Transition", "We offer tips and strategies for making a successful transition to a new career, including researching new industries and building new skills.");  INSERT INTO thread (user\_id, thread\_name, thread\_content) VALUES (3, "The Future of the Job Market", "Lets explore the changing job market, trends in the economy, and how job seekers can prepare for the future of work.");  INSERT INTO thread (user\_id, thread\_name, thread\_content) VALUES (4, "Diversity and Inclusion in the Workplace", "Let's discuss the importance of diversity and inclusion in the workplace and how job seekers can find companies that value diversity.");    b. User  INSERT INTO user (user\_name, user\_email, user\_password, user\_country)  VALUES ("Allan Kay", "allankay69@gmail.com", "ieu4-81ka-1a6d", "Philippines");  INSERT INTO user (user\_name, user\_email, user\_password, user\_country)  VALUES ("Jon Bellic", "jonbellic91@gmail.com", "p245-asi2-aps9", "Serbia");  INSERT INTO user (user\_name, user\_email, user\_password, user\_country)  VALUES ("Emiline Riseborough", "eriseborough0@cbc.ca", "anG94X", "Canada");  INSERT INTO user (user\_name, user\_email, user\_password, user\_country)  VALUES ("Jonell Benit", "jbenit1@eventbrite.com", "RxKnamyZ", "United States");  INSERT INTO user (user\_name, user\_email, user\_password, user\_country)  VALUES ("Reggie Wayper", "rwayper2@plala.or.jp", "DncdKkrAgW", "United States");  INSERT INTO user (user\_name, user\_email, user\_password, user\_country)  VALUES ("Fayina Maliffe", "fmaliffe3@gravatar.com", "Mo8Ke1cAdl", "Ireland");  INSERT INTO user (user\_name, user\_email, user\_password, user\_country)  VALUES ("Anson Howerd", "ahowerd8@prlog.org", "ildpdPNmFZq", "United States");  INSERT INTO user (user\_name, user\_email, user\_password, user\_country)  VALUES ("Jean Reyes", "jreyes96@gmail.com", "f12s-0svk-9suh", "Philippines");  INSERT INTO user (user\_name, user\_email, user\_password, user\_country)  VALUES ("Marianne Seed", "mseed49@gmail.com", "ipsf-96ka-52aa", "United States");  INSERT INTO user (user\_name, user\_email, user\_password, user\_country)  VALUES ("Joshua Allen", "jallen99@gmail.com", "iss3-951a-sjx2", "Philippines");    c. Messages  INSERT INTO message (`#sender\_id`, `#receipient\_id`, msg\_time\_date, msg\_content)  VALUES (4, 3, '2022-12-15 16:05:11', "Hi! I would like to ask permission from you if I can fork your Earthquake Detector Project?");  INSERT INTO message (`#sender\_id`, `#receipient\_id`, msg\_time\_date, msg\_content)  VALUES (2, 5, '2022-12-11 09:13:11', "Hi! I would like to inquire about any hiring offers currently available for the Autonomous Engineering Field?");  INSERT INTO message (`#sender\_id`, `#receipient\_id`, msg\_time\_date, msg\_content)  VALUES (4, 5, '2022-12-20 11:12:24', "Hi! I would like to ask about the deal between Tesla and Google partnership?");  INSERT INTO message (`#sender\_id`, `#receipient\_id`, msg\_time\_date, msg\_content)  VALUES (4, 5, '2022-12-24 10:34:15', "I would like to ask for permission if I could fork one of your projects?");  INSERT INTO message (`#sender\_id`, `#receipient\_id`, msg\_time\_date, msg\_content)  VALUES (1, 5, '2022-12-25 15:11:19', "Hi! I would like to work with your company? When will you guys be hiring?");  INSERT INTO message (`#sender\_id`, `#receipient\_id`, msg\_time\_date, msg\_content)  VALUES (3, 2, '2022-12-30 16:17:23', "Hi! We would like to utlize the CUDA architecture onto our new Azure systems.");  INSERT INTO message (`#sender\_id`, `#receipient\_id`, msg\_time\_date, msg\_content)  VALUES (4, 3, '2022-12-15 16:05:11', "Hi! I would like to ask permission from you if I can use your study for my own paper?");  INSERT INTO message (`#sender\_id`, `#receipient\_id`, msg\_time\_date, msg\_content)  VALUES (7, 4, '2022-12-17 19:35:52', "Hi! I would like to ask permission from you to be one of the oral defense judges for my study?");  INSERT INTO message (`#sender\_id`, `#receipient\_id`, msg\_time\_date, msg\_content)  VALUES (8, 6, '2022-12-19 17:03:22', "Hi! I would like to ask permission from you to judge my study?");  INSERT INTO message (`#sender\_id`, `#receipient\_id`, msg\_time\_date, msg\_content)  VALUES (4, 9, '2022-12-21 20:15:41', "Hi! I would like to ask permission from you to critique my study?");  d. Bulk Mail  INSERT INTO bulk\_mail (`#admin\_id`, send\_date, mail\_subject, mail\_content)  VALUES (1, '2021-07-04', "Happy Independence Day!", "Happy Independence Day for everyone! - From the ABC Jobs Team");  INSERT INTO bulk\_mail (`#admin\_id`, send\_date, mail\_subject, mail\_content)  VALUES (1, '2022-04-17', "Anniversary!", "Happy 5 years anniversary to the ABC Jobs Community Portal site!");  INSERT INTO bulk\_mail (`#admin\_id`, send\_date, mail\_subject, mail\_content)  VALUES (1, '2022-05-30', "Memorial Day!", "Happy Memorial Day from ABC Jobs!");  INSERT INTO bulk\_mail (`#admin\_id`, send\_date, mail\_subject, mail\_content)  VALUES (1, '2022-09-05', "Labor Day!", "Happy Labor Day to all! – From ABC Jobs");  INSERT INTO bulk\_mail (`#admin\_id`, send\_date, mail\_subject, mail\_content)  VALUES (1, '2022-10-10', "Happy Columbus Day!", "Happy Columbus Day to all! – From ABC Jobs");  INSERT INTO bulk\_mail (`#admin\_id`, send\_date, mail\_subject, mail\_content) VALUES (1, '2022-11-11', "Happy Veterans Day!", "Happy Veterans Day to all! – From ABC Jobs");  INSERT INTO bulk\_mail (`#admin\_id`, send\_date, mail\_subject, mail\_content)  VALUES (1, '2022-11-24', "Happy Thanksgiving Day!", "Happy Thanksgiving Day to all! – From ABC Jobs");  INSERT INTO bulk\_mail (`#admin\_id`, send\_date, mail\_subject, mail\_content)  VALUES (1, '2022-12-25', "Happy Christmas Day!", "Happy Christmas Day to everyone! – From ABC Jobs");  INSERT INTO bulk\_mail (`#admin\_id`, send\_date, mail\_subject, mail\_content)  VALUES (1, '2023-01-01', "Happy New Years!", "Happy New Years to all! – From ABC Jobs");  INSERT INTO bulk\_mail (`#admin\_id`, send\_date, mail\_subject, mail\_content)  VALUES (1, '2023-08-25', "Happy National Independence Day!", "Happy Independence Day to all Filipinos! – From ABC Jobs");    e. Jobs INSERT INTO jobs (`#user\_id`, job\_name, job\_companyname, job\_address, job\_salary)  VALUES (1, "Junior Software Engineering", "Apple Inc.", "California, United States", 40000);  INSERT INTO jobs (`#user\_id`, job\_name, job\_companyname, job\_address, job\_salary)  VALUES (2, "Senior Software Engineering", "NVIDIA", "California, United States", 200000);  INSERT INTO jobs (`#user\_id`, job\_name, job\_companyname, job\_address, job\_salary)  VALUES (3, "Junior Software Engineering", "Microsoft", "Washington D.C., United States", 38000);  INSERT INTO jobs (`#user\_id`, job\_name, job\_companyname, job\_address, job\_salary)  VALUES (4, "Senior Software Engineering", "Google", "California, United States", 230000);  INSERT INTO jobs (`#user\_id`, job\_name, job\_companyname, job\_address, job\_salary)  VALUES (5, "Senior Software Engineering", "Tesla", "Texas, United States", 100000);  INSERT INTO jobs (`#user\_id`, job\_name, job\_companyname, job\_address, job\_salary)  VALUES (6, "Junior Software Engineering", "Intel Corporation", "California, United States", 40000);  INSERT INTO jobs (`#user\_id`, job\_name, job\_companyname, job\_address, job\_salary)  VALUES (7, "Junior Software Engineering", "Oracle Corporation", "Austin, Texas, US", 60000);  INSERT INTO jobs (`#user\_id`, job\_name, job\_companyname, job\_address, job\_salary)  VALUES (8, "Senior Software Engineering", "AMD", "California, United States", 80000);  INSERT INTO jobs (`#user\_id`, job\_name, job\_companyname, job\_address, job\_salary)  VALUES (9, "Junior Software Engineering", "Floatplane Inc", "Vancouver, BC, Canada", 30000);  INSERT INTO jobs (`#user\_id`, job\_name, job\_companyname, job\_address, job\_salary)  VALUES (10, "Senior Software Engineering", "Apple Inc", "California, United States", 80000);    f. Work Experience  INSERT INTO experience (prev\_job, company\_name, date\_start, date\_end, year\_count, `#user\_id`)  VALUES ("iOS App Developer", "Google", '2019-11-15', '2021-11-20', 1, 1);  INSERT INTO experience (prev\_job, company\_name, date\_start, date\_end, year\_count, `#user\_id`)  VALUES ("Junior Software Developer", "NVIDIA", '2017-07-11', '2019-07-08', 1, 2);  INSERT INTO experience (prev\_job, company\_name, date\_start, date\_end, year\_count, `#user\_id`)  VALUES ("iOS App Development", "Google", '2016-05-07', '2020-11-19', 4, 3);  INSERT INTO experience (prev\_job, company\_name, date\_start, date\_end, year\_count, `#user\_id`)  VALUES ("Junior Software Developer", "Google", '2014-06-12', '2018-07-19', 4, 4);  INSERT INTO experience (prev\_job, company\_name, date\_start, date\_end, year\_count, `#user\_id`)  VALUES ("Leader, Machine Learning Team", "Tesla", '2013-02-11', '2016-04-19', 3, 5);  INSERT INTO experience (prev\_job, company\_name, date\_start, date\_end, year\_count, `#user\_id`)  VALUES ("Android App Development", "Microsoft", '2017-06-11', '2019-09-17', 2, 6);  INSERT INTO experience (prev\_job, company\_name, date\_start, date\_end, year\_count, `#user\_id`)  VALUES ("Java Developer", "Google", '2013-04-21', '2018-09-12', 5, 7);  INSERT INTO experience (prev\_job, company\_name, date\_start, date\_end, year\_count, `#user\_id`)  VALUES ("React Native Developer", "Discord", '2019-11-16', '2021-06-11', 1, 8);  INSERT INTO experience (prev\_job, company\_name, date\_start, date\_end, year\_count, `#user\_id`)  VALUES ("HTML Web Developer", "Google", '2012-06-11', '2021-11-19', 9, 9);  INSERT INTO experience (prev\_job, company\_name, date\_start, date\_end, year\_count, `#user\_id`)  VALUES ("Human Interface Development", "Apple Inc", '2015-07-11', '2020-07-19', 5, 10);     1. Design report and Create queries which will be used for generation of meaningful management reports. Present the report in HTML format (1 report)  |  |  |  |  | | --- | --- | --- | --- | | No | Note | Query | Evidence | | 1 | Recent Messages | SELECT \* FROM communityportal.message ORDER BY msg\_time\_date DESC LIMIT 10; | Show 10 recent messages in the database. | | 2 | Fetch all jobs. | SELECT \* FROM communityportal.jobs; | Show all the jobs entered in the database. | | 3 | Fetch all user’s experience. | SELECT communityportal.user.`#user\_id`, communityportal.user.user\_name, communityportal.experience.prev\_job, communityportal.experience.year\_count FROM communityportal.user JOIN communityportal.experience ON communityportal.user.`#user\_id` = communityportal.experience.`#user\_id`; | Shows all of the user’s job experience in the database. | | 4 | Fetch all user’s email and country info. | SELECT communityportal.user.`#user\_id`, communityportal.user.user\_name, communityportal.user.user\_email, communityportal.user.user\_country FROM communityportal.user; | Shows all the email and country info from each user in the database. | | 5 | Fetch user’s threads. | SELECT communityportal.user.`#user\_id`, communityportal.user.user\_name, communityportal.thread.thread\_name, communityportal.thread.thread\_dateTime, communityportal.thread.thread\_content FROM communityportal.user JOIN communityportal.thread ON communityportal.user.`#user\_id` = communityportal.thread.`#user\_id`; | Shows all the threads created by each user. |   Report Recent Messages Log  Report All Jobs Log  Report All User’s Experience Log  Report All Users Email and Country info  Report All User’s Threads |
|  |

1. Discuss briefly test methods you will employ to test and validate the database and brief reasons why you choose each test.

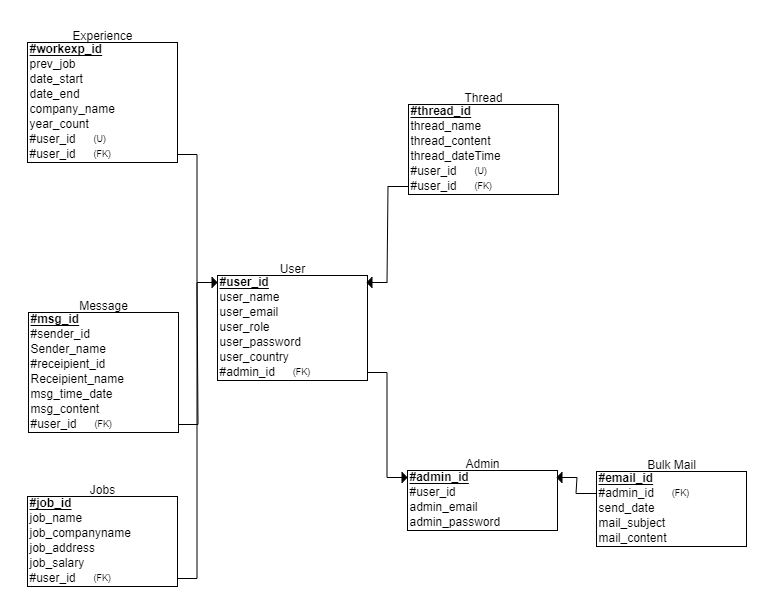
3.1. Structural Database Testing

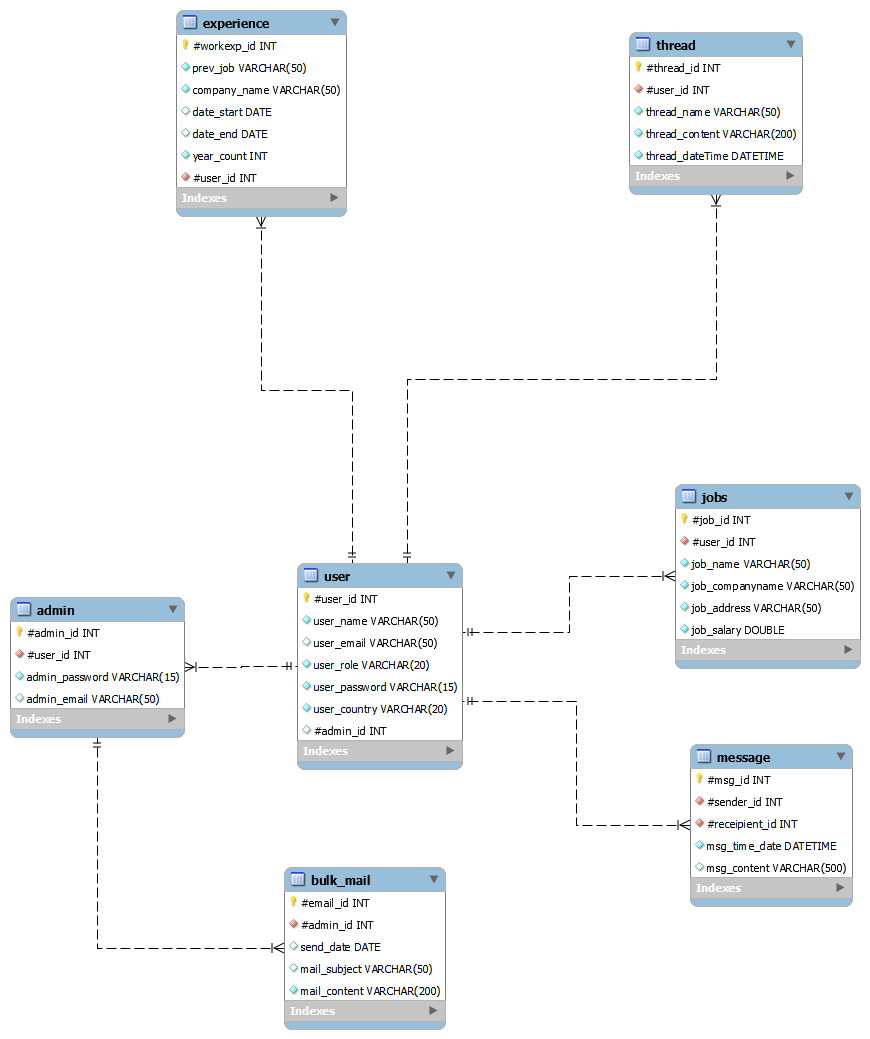
Schema Testing

* This is used to test the database’s structure and integrity.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. / Test Cases | Expected Result | Actual Result | Test Result | Evidence |
| 1. Schema Testing | The Relationship Schema and EER Diagram should be similar. | The Relationship Schema and the EER Diagram are similar. | Pass | Screen Capture of the Relationship Schema and EER Diagram below. |

Evidence:

Relationship Schema

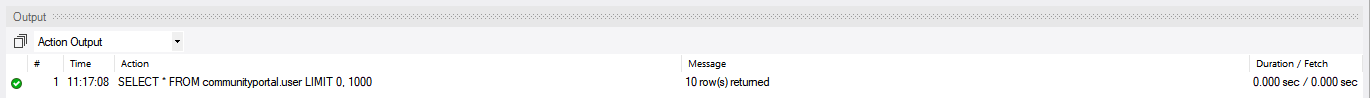
EER Diagram

3.2. Table / Columns Testing

- The purpose of Table/Column testing is to verify if the table can handle the entry of a wide variety of data, variables and scenarios.

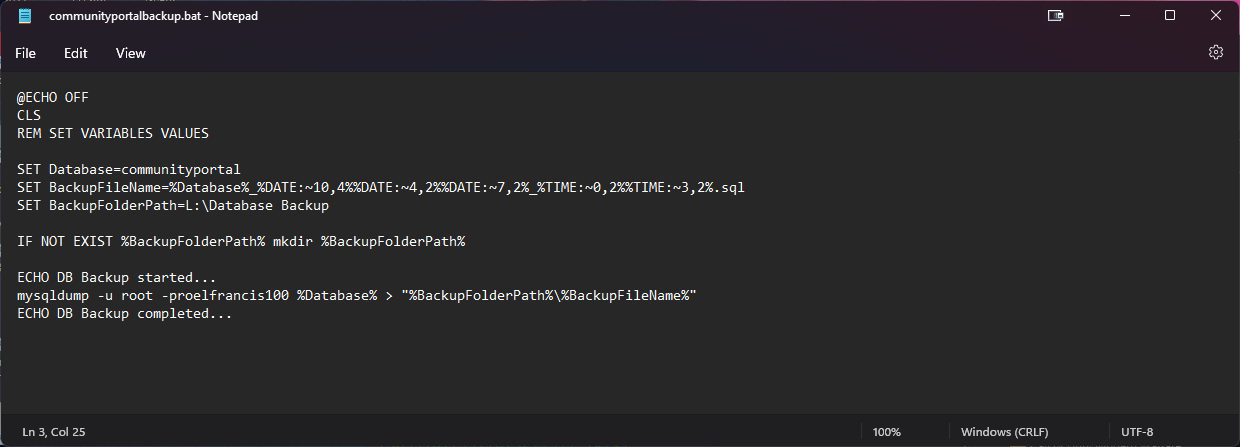
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TestCaseID | Test Case | Expected Result | Actual Result | Pass / Fail | Evidence |
| TC01 | Table Existence Test: Add data to the ‘user’ table to verify if exists or not. | Logs should show ’10 row(s) returned’ | Logs reported show ’10 row(s) returned’ | Pass | Screen Capture provided below. |
| TC02 | Data Insertion Test: Check if the table is able to receive and enter 1 data entry. | Logs should report “1 row(s) affected” | Logs reported “1 row(s) affected” | Pass | Screen Capture provided below. |
| TC03 | Primary Key Test: Insert a duplicate value for the #thread\_id column for thread. | Logs should report an error indicating a Duplicate entry for thread ‘PRIMARY’. | Logs reported an error indicating a Duplicate entry for thread ‘PRIMARY’. | Pass | Screen Capture provided below. |
| TC04 | Data Type Test: Insert a email that has more than 50 characters on the user\_email column. | Log should report an error indicating tha the “Data too long for column ‘user\_email’. | Logs reported an error “Data too long for column ‘user\_email’. | Pass | Screen Capture provided below. |
| TC05 | Column Existence Test: Check if the column `msg\_content` is available. | Logs should report “10 row(s) returned”. | Log’s reported “10 row(s) returned” | Pass | Screen Capture provided below. |

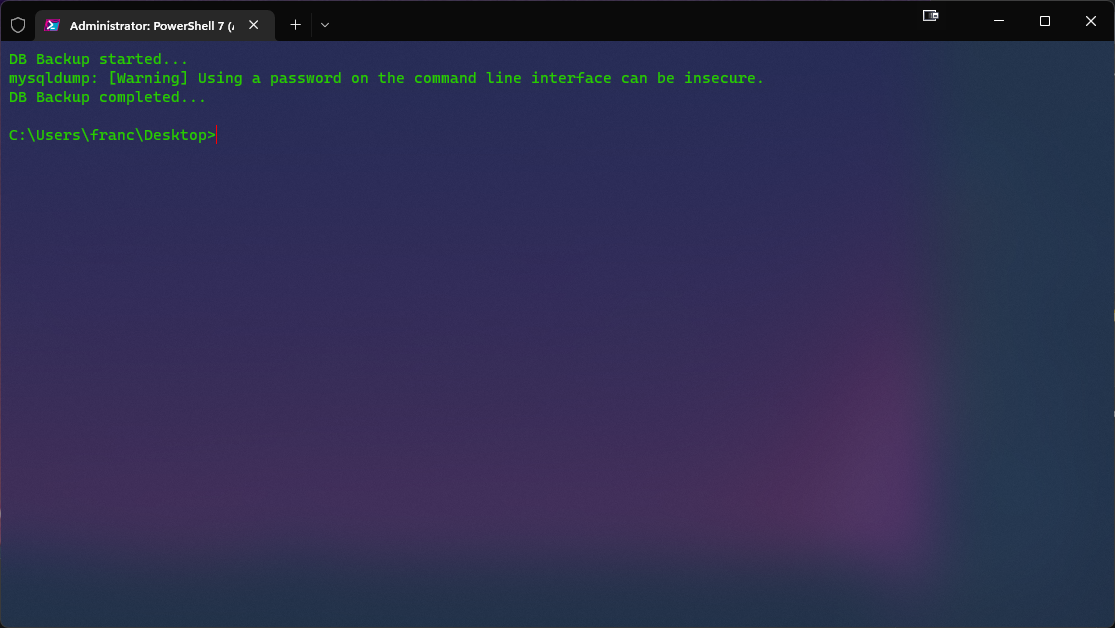
Screen Capture:

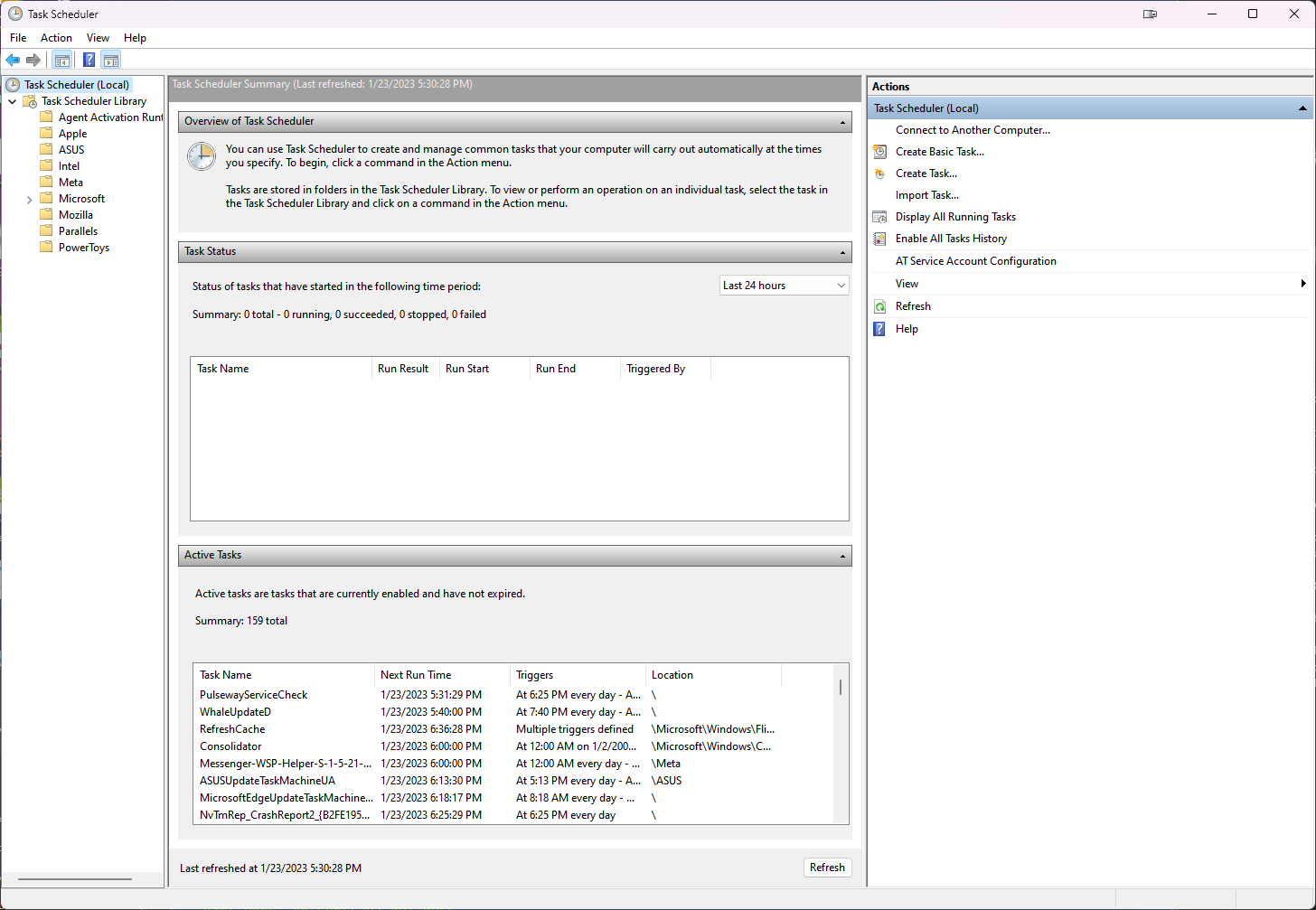
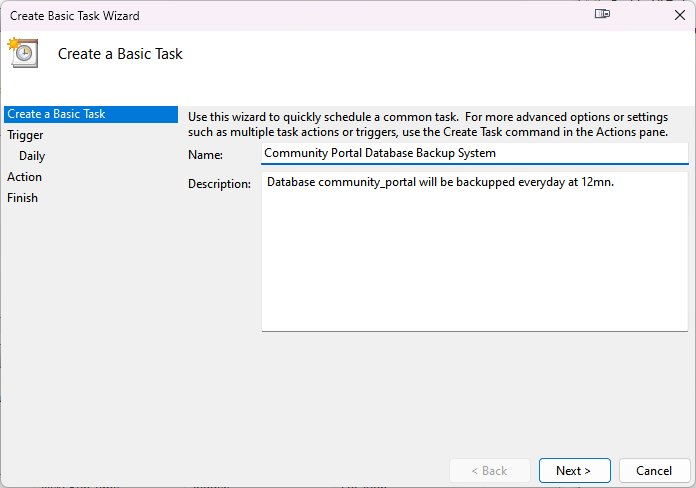
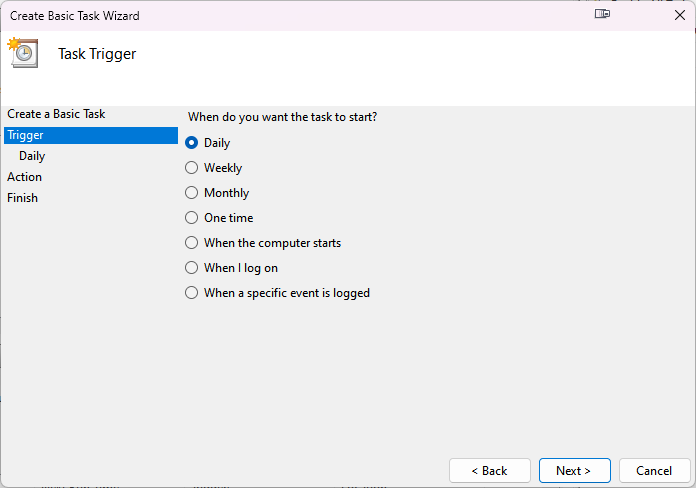
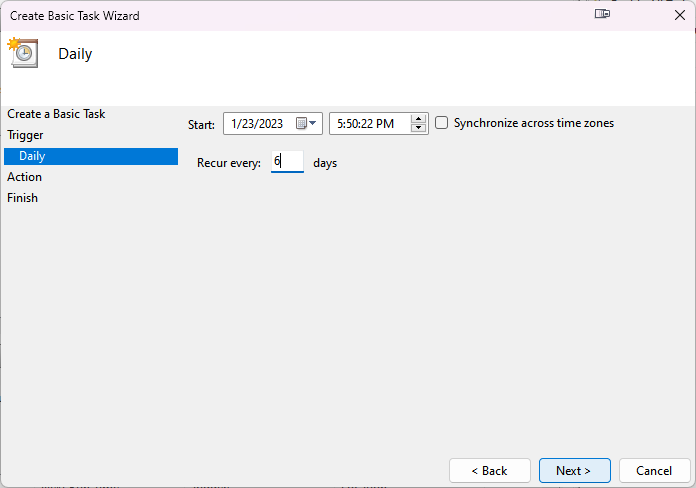
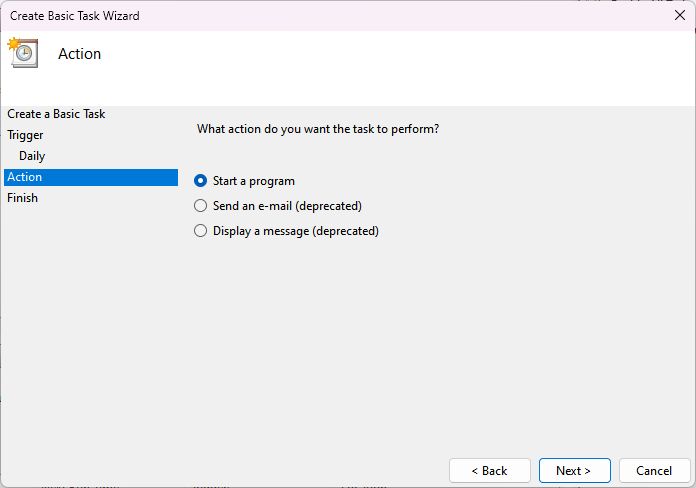
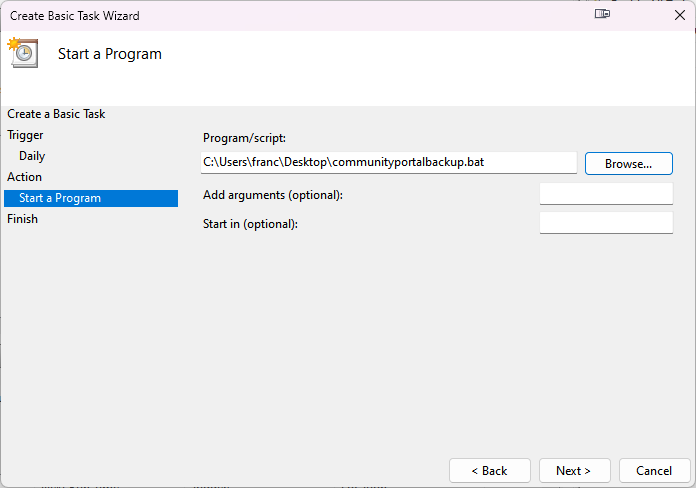
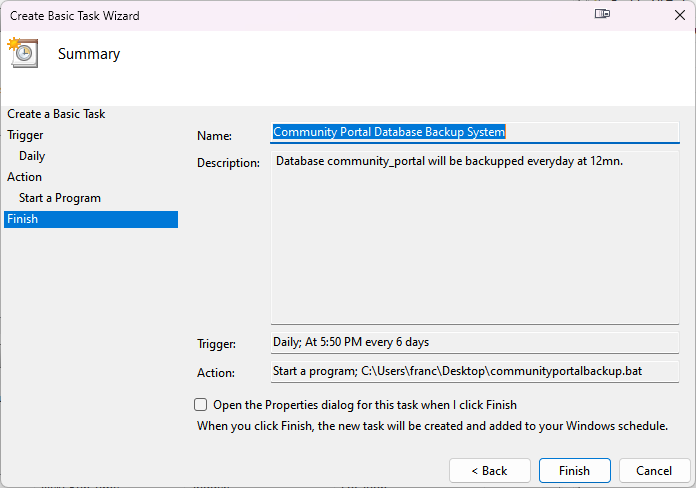
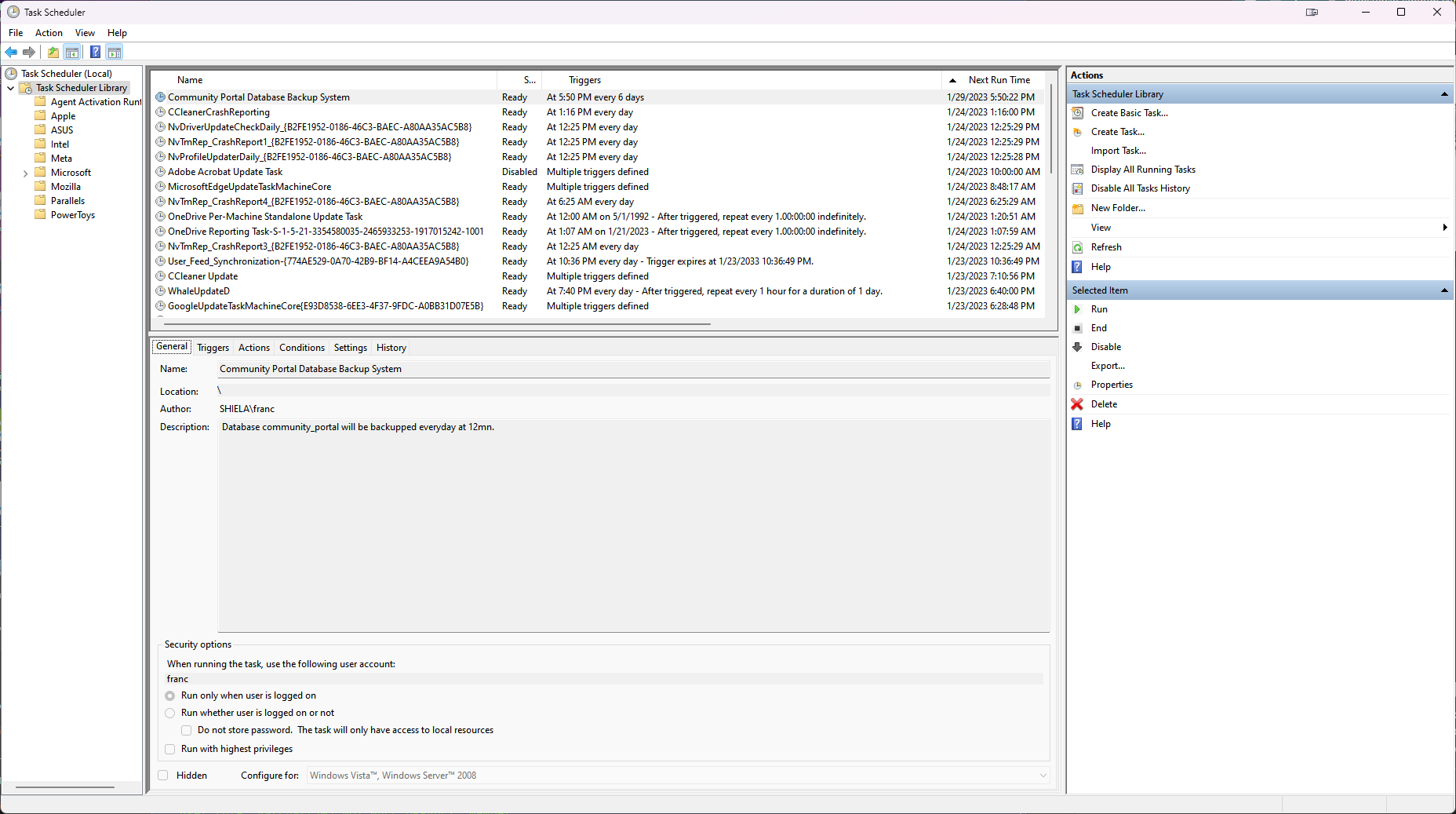
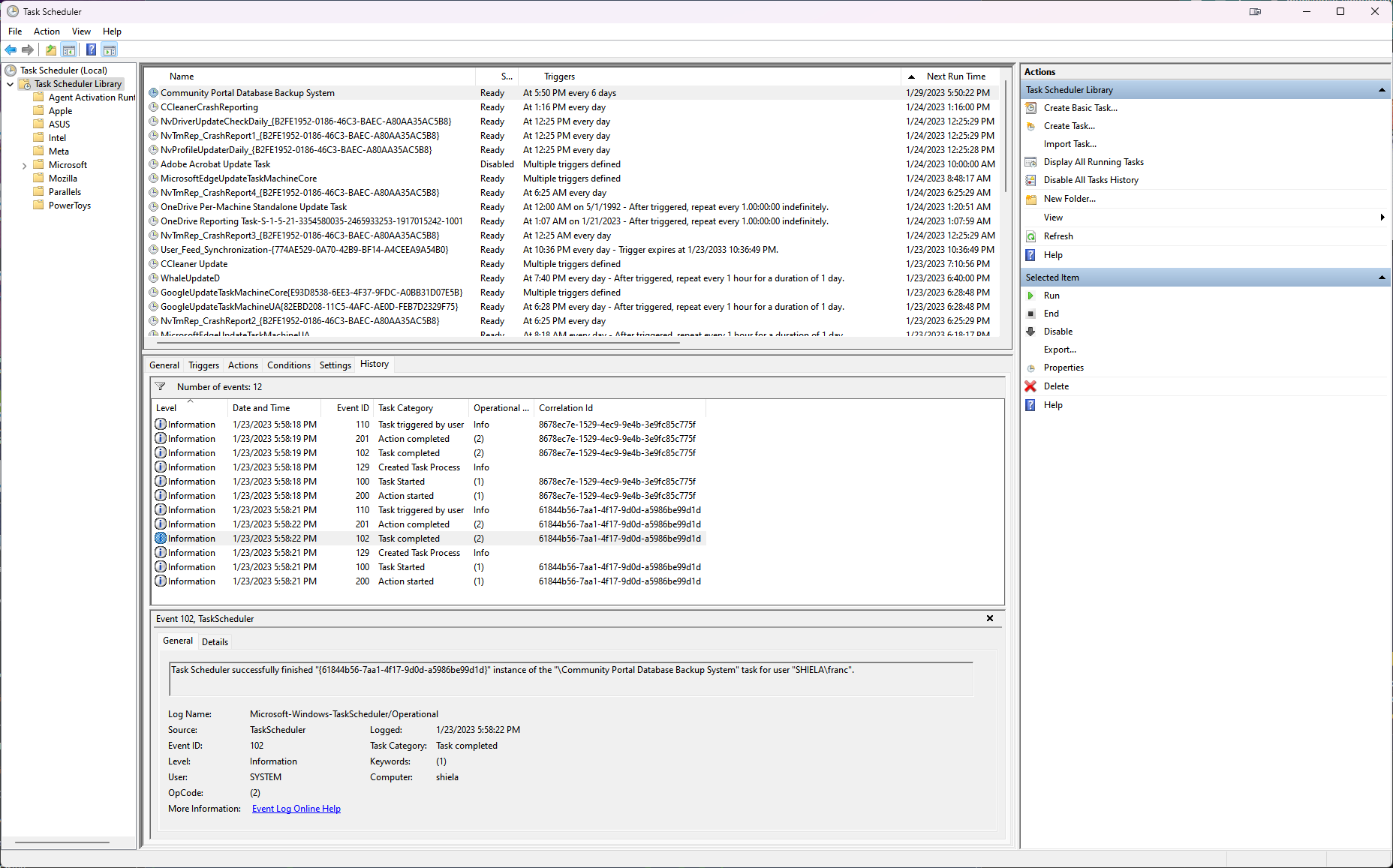
1. Test #1  
   SELECT \* FROM communityportal.user;
2. Test #2  
   INSERT INTO communityportal.user (user\_name, user\_email, user\_role, user\_password, user\_country)  
   VALUES ("Adam Phil", "adamphil21@gmail.com", "member", "ads82-ad9a-9saw", "United States");  
   
3. INSERT INTO communityportal.thread (`#thread\_id`, `#user\_id`, thread\_name, thread\_content, thread\_dateTime)  
   VALUES (1, 5, "Testing", "Testing123", "2023-01-21 11:59:57");  
   
4. INSERT INTO communityportal.user (user\_email)  
   VALUES ("qwertyreeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee@gmail.com");  
   
5. SELECT msg\_content FROM communityportal.message;  
   
6. Document the database



1. Create a batch script to backup database & schedule it to run every 6 hours using windows task scheduler. Provide the script, along with the screen capture of Windows Task scheduler.
2. Create and run the backup batch file





1. Configure Task Scheduler
2. Open Task Scheduler  
   
3. Click on Create Basic Task then add your name and the description of the task  
   
4. Set start trigger for the task  
   
5. Set the start date of the task and its interval.  
   
6. Set the task to start a program  
   
7. Select your backup script  
   
8. Review your configuration then click finish  
   
9. Navigate to Task Scheduler Library on the left side then check your task  
     
   
10. Provide a Restoration script in case of failure

