|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PYTHON/MySQL PROJECT CHALLENGE RUBRIC** | | | | | | | | | | | | |
|  | **Database Principles using** | | | | | | | | | | **Possible Mark** | **Mark Obtained** |
|  | **Understanding problem as reflected by the design structures** | | | | | | | | | |  |  |
|  | **Create a database with an appropriate name.** | | | | | | | | | |  |  |
| 2 | | | | | 2 | | | | 0 | **2** | 2 |
| Database Created with appropriate name | | | | | Database created but name inappropriate | | | | No database created |  |  |
| Create the tables and add **the fields** | | | | | | | | | | | |
| 8 | | 6 | | | 4 | | | | 2 | **8** | 8 |
| All required tables and fields appropriate | | Only 3/4 of the tables created with appropriate fields | | | Half of the tables created with appropriate field sizes | | | | Only1/4 of tables created |  |  |
| **Use of relationships** | | | | | | | | | |  |  |
| 8 | | 6 | | | 4 | | | | 2 | **8** | 8 |
| All primary keys and foreign keys identified and linked correctly | | Tables created with keys incorrectly linked causing some errors | | | Tables created with only primary keys and more a relationship | | | | Tables created but no relationships. Only primary keys set up |  |  |
| **Database Backup** | | | | | | | | | | **2** | 2 |
| Database backup submitted together with the task | | | | | Database not submitted | | | | |  |  |
| 2 | | | | | 0 | | | | |  |  |
|  | | | | | | | | | |  |  |
| **SQL STATEMENTS** | | | | | | | | | |  |  |
| Advanced and appropriate SQL code implemented | Three quarters of SQL code work and produce desired results | | | | | | Half of the SQL code implemented and produce | Less than half of the SQL code implemented | |  |  |
| 8 | 6 | | | | | | 4 | 2 | | **8** | 6 |
|  | | | | | | | | | | | |
| Command Line Interface-Readability and communicates clearly with user on input request and out display | | | | | | | | | |  |  |
| Advanced | Skilled | | | | | | Intermediate | Basic | | **8** | 7 |
| 8 | 6 | | | | | | 4 | 2 | |  |  |
| **Connection to the database** | | | | | | | | | |  |  |
| Correct connection to the database with MySQL module imported and cursor | Connection to the database with MySQL module imported but has errors | | | | | | Database connection partially complete OR cursor not declared | Incorrect declaration of connection or cursor | |  |  |
| 8 | 6 | | | | | | 4 | 2 | | **8** | 8 |
| **Use of iteration structures. (For loop, Do While, )** | | | | | | | | | |  |  |
| 8 | | 6 | | | 4 | | | | 2 | **8** | 6 |
| Used appropriate and most effective repetition structures to solve the problem in all instances | | Used appropriate and most effective repetition structures in most instances | | | Appropriate and most effective use of repetition structures in less than 50% of the instances | | | | Inappropriate or ineffective use of repetition structures |  |  |
| **Use of Functions, (parameter passing) including in-built functions** | | | | | | | | | |  |  |
| 8 | | 6 | | | 4 | | | | 2 | **8** | 6 |
| Excellent interaction/ communication between modules/classes. Includes parameter passing | | Proficient/appropriate use of modules/functions or other sub-procedures with small flaws. Includes parameter passing between modules/ functions or other sub-procedures but not always appropriate or correct. | | | Limited use of modules/functions or other sub-procedures. Limited parameter passing | | | | No use of modules/functions or other sub-procedures. No parameters passed |  |  |
| **Complexity of Code** | | | | | | | | | |  |  |
| 8 | | 6 | | | 4 | | | | 2 | **8** | 6 |
| Advanced. Other techniques such as OOP implemented and testing | | Skilled  Some sections of code produce errors. Some testing attempted. | | | Intermediate  Average skills implemented | | | | Basic Very limited skills learnt in the course have been implemented |  |  |
|  | **Error/Exception Handling** | | | | | | | | | |  |  |
| 8 | | 6 | | | 4 | | | | 2 | **8** | 7 |
| Excellent exception handling by the use of selection structures and try, catch statements. Proper Feedback given to users. | | Exceptions minimized by the use of selection structures and try, catch statements. No Feedback given to user | | | Exception minimized by the use of selection structures only. No Feedback given to the user. | | | | Attempt to implement exception resulted in logic or syntax errors |  |  |
| **Program Compiles Successfully** | | | | | | | | | |  |  |
| 8 | | 6 | | | 4 | | | | 2 | **8** | 7 |
| No run time or syntax errors. All the options are executed successfully | | Program produce runtime errors only | | | Some of the options produce errors when executed | | | | Majority of the code produces errors. No evidence of tests |  |  |
| **Output** | | | | | | | | | |  |  |
| 6 | | | | | | 4 | | | 2 | **6** | 5 |
| Output correct Matches the user requirements | | | | | Some sections produce incorrect output | | | | Incorrect logic. Not meeting user requirements |  |  |
| **Program Documentation** | | | | | | | | | |  |  |
| 8 | | 6 | | | 4 | | | | 2 | **8** | 8 |
| Code well documented in all paths | | 75% of code documented | | | Only 50%of code is documented | | | | Just 20% of the code is documented |  | Well Done |
| **General Evaluation** | **Time Management: (0 = always late, work was never done)** | | | | | | | | | |  |  |
| 8 | | | | 6 | | | | | 4 | **8** | 8 |
| All deadlines met | | | Project submitted few hours after due date | | | | | | Project submitted after interventions |  |  |
|  | | | | | | | | | | **106** | 94 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FINAL COMPETENCE LEVEL INDICATORS** | | | | | |
| **Outstanding** | **Highly competent** | | **Competent** | **Not yet competent** | **Not achieved** |
| **5** | **4** | | **3** | **2** | **1** |
| **Student’s Competence Level:** | | **PASS 89%** | | | |
| **Student’s Name and Signature:** | | Josh | | | |
| **Assessor: Thapelo Tsotetsi** | |  | | | |
| **Date:** | |  | | | |
| Well done for providing reports and readme files as evidence of how the development progressed. There is solid evidence of planning and tests as reflected by the existence of test data in the tables. Self-explanatory menu. Good work. After registering a user in admin screen- the program must direct user to main menu whether he/she want to register again. I did not see grant and revoke options. Loops and error handling well executed in 75% of the code. A for admin though its hidden as 4. Well done on documentation. | | | | | |