Expense Tracker Application – Report

The following table is a brief report of how each of the following tasks fits into the Expense Tracker as well as it’s location within the project.

|  |  |  |  |
| --- | --- | --- | --- |
| Task | File Location | Line # | Task Achieved |
| The project should be written using Python and MySQL and should use Flask with Flask-MySQLAlchemy. |  |  |  |
| You may add css and/or javascript |  |  |  |
| Include at least one structural change to the database using |  |  |  |
| Provide the DDL as well as the INSERT SQL for creating the tables and initially populating the database |  |  |  |
| Include at least one insertion of a new record that will occur during the execution of the application. |  |  |  |
| Include at least one update of a record--changing an existing record |  |  |  |
| Include at least one delete of a record.  Use SQLAlchemy. |  |  |  |
| Include at least one simple SELECT SQL statement.  Use regular SQL for this. |  |  |  |
| Include one query using Flask-SQLAlchemy filter or filter\_by. |  |  |  |
| Include at least one SELECT using an aggregate function |  |  |  |
| Include at least one SELECT using a compound condition using regular SQL, and also the equivalent of a compound condition select using Flask-SQLAlchemy. |  |  |  |
| Include at least one JOIN query using SQL, and also one using Flask-SQLAlchemy. |  |  |  |
| Include at least one subquery.  Regular SQL.  Excellence points if you also use Flask-SQLAlchemy. |  |  |  |
| Use a form to collect user data, as shown in our CRUD labs. |  |  |  |
| Populate a field on a form or table *from* the database. |  |  |  |
| Check for empty data fields. You can use the built-in validations for this. |  |  |  |
| Implement referential integrity |  |  |  |
| Use an appropriate structure for your project package. |  |  |  |