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 its location within the project. All files are within the /flaskDemo/ folder and further sub directories will be noted

|  |  |  |  |
| --- | --- | --- | --- |
| Task | File Location | Line # | Task Achieved |
| The project should be written using Python and MySQL and should use Flask with Flask-MySQLAlchemy. | forms.py  models.py  routes.py | Entire File | Allows connection between database, web application functionality, and deployment |
| You may add css and/or javascript | -N/A | -N/A | -N/A |
| Include at least one structural change to the database | forms.py | Line 144 | form used to create a payment |
| Include at least one insertion of a new record that will occur during the execution of the application. | forms.py | Line 141 | form used to create a budget |
| Include at least one update of a record--changing an existing record | forms.py | Line 127 | form used to update budget |
| Include at least one delete of a record.  Use SQLAlchemy. | routes.py | Line 229 | Route for deleting a budget |
| Include at least one simple SELECT SQL statement.  Use regular SQL for this. | routes.py | Line 223 | Route for selecting a budget |
| Include one query using Flask-SQLAlchemy filter or filter\_by. | routes.py | Line 141 | Queries the database to username to ensure the correct username is used. |
| 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. | Routes.py | Line 110 | Ensures payments belong to the currently logged in user |
| Include at least one JOIN query using SQL, and also one using Flask-SQLAlchemy. | routes.py | Line 190 | Joins Budget and Payment over the same expense ID |
| 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. | forms.py | Line 69 | form used to accept login information |
| Populate a field on a form or table *from* the database. | forms.py | Line 127 | form used to update budget |
| Check for empty data fields. You can use the built-in validations for this. | forms.py | Line 128 | form used to update budget name mandating data and length <=15 char |
| Implement referential integrity |  |  |  |
| Use an appropriate structure for your project package. | Refers to entire directory within /flaskdemo/ |  |  |