```
-app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///db/Student.db'
+app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///Student.db'
 app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False

 # Initialize SQLAlchemy
 db = SQLAlchemy(app)

+class Student(db.Model):
+    id = db.Column(db.Integer, primary_key=True)
+    name = db.Column(db.String(100), nullable=False)
+    solution = db.Column(db.Text, nullable=True)
+    grade = db.Column(db.String(10), nullable=False)
+
+    def __repr__(self):
+        return f'<Student {self.name}>'
+
+# Create tables if not already created
+migrate = Migrate(app, db)
+
+with app.app_context():
+    db.create_all()
+
+
+
 @app.route("/")
 def home():
     return render_template("index.html")
@@ -45,21 +63,66 @@ def grading_dashboard():
         {"input": "[1, -2, -4, 3, 0]", "expected": "-6"},
```

```
      {"input": "[100, 101, 102]", "expected": "202"}
    ]
+   students = Student.query.all()
+   print(students)
+   student_data = []
+
+   for student in students:
+       student_data.append({
+           "name": student.name,
+           "solution": student.solution or "No solution provided",
+           "grade": student.grade
+       })
+
+   return render_template("grading_dashboard.html", assignment=assignment,
test_cases=test_cases, students=student_data)
+@app.route('/delete')
+def delete_students():
+   # Delete all students
+   Student.query.delete()
+   db.session.commit()
+
+   return "All students deleted from the database!"
+
+@app.route('/add_students')
+def add_students():
+   # Insert new students if not already inserted
+   if not Student.query.first():  # Only insert if the table is empty
+       student1 = Student(name='AliceHa', solution='def sum_even(numbers):\n    return sum(x for x
in numbers if x % 2 == 0)', grade='A')
```

```
+      student2 = Student(name='Bob', solution='def sum_even(numbers):\n    return sum(x for x in numbers if x % 2 == 0)', grade='B')
+      student3 = Student(name='Charlie', solution='def sum_even(numbers):\n    return sum(x for x in numbers if x % 2 == 0)', grade='A')
+      student4 = Student(name='Diana', solution='def sum_even(numbers):\n    return sum(x for x in numbers if x % 2 == 0)', grade='C')
+
+      db.session.add_all([student1, student2, student3, student4])
+      db.session.commit()
+
+      return "Students added to the database!"
+   else:
+      return "Students already exist in the database!"
+
+
+@app.route('/test_connection')
+def test_connection():
+   try:
+      # Execute a simple query to test connection
+      result = db.session.execute(text('SELECT 1')).fetchone()
+
+      # Check if query result is valid
+      if result and result[0] == 1:
+         return "Connection to the database is successful!"
+      else:
+         return "Failed to connect to the database."
+   except Exception as e:
+      return f"Error connecting to the database: {str(e)}"
+
+@app.route('/view_students')
```

+def view_students():

+ # Query all students

+ students = Student.query.all()

+

+ # Return a simple HTML response showing the students

+ return render_template('students_list.html', students=students)

- students = [

- {"name": "Alice", "solution": "def sum_even(numbers):\n return sum(x for x in numbers if x % 2 == 0)", "correct": True, "annotation": "Correct solution."},

- {"name": "Bob", "solution": "def sum_even(numbers):\n total = 0\n for x in numbers:\n if x % 2 == 1:\n total += x\n return total", "correct": False, "annotation": "Checks for odd numbers instead of even."},

- {"name": "Charlie", "solution": "def sum_even(numbers):\n total = 0\n for num in numbers:\n if num % 2 == 0:\n total += num\n return total", "correct": True, "annotation": "Correct solution."},

- {"name": "Diana", "solution": "def sum_even(numbers) # Missing colon here\n total = 0\n for x in numbers:\n if x % 2 == 0:\n total += x\n return total", "correct": False, "annotation": "Syntax error: Missing colon in function definition."},

- {"name": "Evan", "solution": "def sum_even(numbers):\n count = 0\n for num in numbers:\n if num % 2 == 0:\n count += 1 # Wrong: should add the number, not count it.\n return count", "correct": False, "annotation": "Counts even numbers instead of summing their values."},

- {"name": "Fiona", "solution": "def sum_even(numbers):\n total = 0\n for num in numbers:\n if num > 0 and num % 2 == 0: # Fa

ulty: Excludes negative even numbers.\n total += num\n return total", "correct": False, "annotation": "Only sums positive even numbe

rs, ignoring negatives."},

- {"name": "George", "solution": "def sum_even(numbers):\n return sum(numbers) # No condition; adds every number.", "correct": False, "annotation": "Sums all numbers regardless of even/odd."},

- {"name": "Helen", "solution": "def sum_even(numbers):\n even_numbers = []\n for num in numbers:\n if num % 2 == 0:\n even_numbers.append(num)\n result = 0\n for even in even_numbers:\n result += even\n return result", "correct": True, "annotation": "Correct solution (verbose but clear)."},

- {"name": "Ian", "solution": "def sum_even(numbers):\n if not numbers:\n return 0\n head = numbers[0]\n tail = numbers[1:]\n if head % 2 == 0:\n return head + sum_even(tail)\n else:\n return sum_even(tail)", "correct": True, "annotation": "Correct recursive solution."},

- {"name": "Julia", "solution": "def sum_even(numbers):\n total = 0\n for num in numbers:\n # This condition is intended to check evenness but is flawed.\n if (num / 2) % 2 == 0:\n total += num\n return total", "correct": False, "annotation": "Faulty

condition using division leads to wrong results."}

-]

- return render_template("grading_dashboard.html", assignment=assignment, test_cases=test_cases, students=students)

@app.route("/query", methods=['GET', 'POST'])

def query():