Phase 4

Phase 4: Dashboard and Reporting
```
   │       │
   │       ├── Build a Simple Dashboard using Flask
   │       │
   │       ├── Connect Dashboard to Database (SQLite/MySQL)
   │       │
   │       └── Generate PDF Reports of Vulnerabilities with Python
```

```
from flask import Flask, render_template, request
from flask_sqlalchemy import SQLAlchemy
from reportlab.lib.pagesizes import letter
from reportlab.pdfgen import canvas

# Initialize Flask app and database
app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///vulnerabilities.db'
db = SQLAlchemy(app)

# Database model for storing vulnerabilities
class Vulnerability(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    scan_name = db.Column(db.String(100), nullable=False)
    status = db.Column(db.String(50), nullable=False)
    severity = db.Column(db.String(50), nullable=False)

# Route to display vulnerability scan results
@app.route('/')
def home():
    vulnerabilities = Vulnerability.query.all()  # Fetch all vulnerabilities from database
    return render_template('index.html', vulnerabilities=vulnerabilities)

# Route to generate PDF report of vulnerabilities
@app.route('/generate_report', methods=["POST"])
def generate_report():
    vulnerabilities = Vulnerability.query.all()  # Get all vulnerabilities from the database
    generate_pdf(vulnerabilities)  # Call function to generate the PDF
    return "Report Generated", 200

# Function to generate a PDF report using ReportLab
def generate_pdf(vulnerabilities):
    c = canvas.Canvas("vulnerability_report.pdf", pagesize=letter)
    c.drawString(100, 750, "Vulnerability Report")
    c.drawString(100, 730, "-------------------------------")
```

```
    y_position = 710

    for vuln in vulnerabilities:
        c.drawString(100, y_position, f"Scan Name: {vuln.scan_name}")
        c.drawString(100, y_position-20, f"Status: {vuln.status}")
        c.drawString(100, y_position-40, f"Severity: {vuln.severity}")
        y_position -= 60

    c.save()  # Save the PDF file

# Initialize the database and create tables
@app.before_first_request
def create_tables():
    db.create_all()  # Create database tables if not already created

# HTML template for displaying vulnerability results (index.html)
@app.route('/index.html')
def index_html():
    return '''


    <!DOCTYPE html>
    <html>
    <head>
        <title>Vulnerability Dashboard</title>
    </head>
    <body>
        <h1>Vulnerability Scan Results</h1>
        <table border="1">
            <tr>
                <th>Scan Name</th>
                <th>Status</th>
                <th>Severity</th>
            </tr>
            {% for vuln in vulnerabilities %}
            <tr>
                <td>{{ vuln.scan_name }}</td>
                <td>{{ vuln.status }}</td>
                <td>{{ vuln.severity }}</td>
            </tr>
            {% endfor %}
        </table>
        <form action="/generate_report" method="POST">
            <button type="submit">Generate PDF Report</button>
        </form>
    </body>
    </html>
    '''
```

```
# Start the Flask web app
if __name__ == "__main__":
    app.run(debug=True)
```